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

Todd Morrill[1] **Aahlad Puli**[2] **Murad Megjhani**[3,4] **Soojin Park**[3,5,6] **Richard Zemel**[1]
[1]Department of Computer Science, Columbia University, USA
[2]Department of Computer Science, New York University, USA
[3]Department of Neurology, Columbia University Medical Center, USA
[4]Department of Computer Science, Barnard College, USA
[5]Department of Biomedical Informatics, Columbia University Medical Center, USA
[6]NewYork-Presbyterian Hospital at Columbia University Medical Center, USA
{todd, zemel}@cs.columbia.edu, aahlad@nyu.edu,
{mm5025, sp3291}@cumc.columbia.edu

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 as calibration error and predictive accuracy. This is due to the restrictive inductive bias mixture-of-experts imposes: 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 discrete-time 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). AI systems are being developed for medical imaging (Erickson et al., 2017), disease diagnosis (Ahsan et al., 2022), and many other clinical support settings. However, AI for CDSS faces barriers to adoption due to clinician mistrust of model predictions (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 interpretability of the model's decision-making process–in this case, the ability to reason by analogy to similar patients. Our work[1] addresses survival analysis, where the task is to predict when clinical events will occur (i.e., time-to-event regression) while contending with right-censoring–not observing the event time for a subset of the patients.

Mixture-of-experts (MoE) models for medical survival analysis are particularly appealing for the above desiderata due to their ability to discover latent groups of patients. MoEs are defined by two key components: (i) a router that assigns patients to groups and (ii) a set of experts that produce event distributions for each group (Nagpal et al., 2021b; Hou et al., 2023; Buginga & e Silva,

[1]See our full work, accepted at the 2025 Machine Learning for Health Symposium, at https://arxiv.org/abs/2511.09567, along with our code at https://github.com/ToddMorrill/survival-moe

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Learning from Time Series for Health.

2024). Our goal is to investigate how expert expressivity in deep, discrete-time MoE heads affects calibration and predictive accuracy under matched capacity, which has not been carefully studied in the medical survival modeling setting. In our study, experts range from fixed prototypes to per-patient parameterizations. We therefore compare three MoE heads that differ only in expert expressivity to isolate its impact and include standard non-MoE baselines for context.

The first architecture, **Fixed MoE**, uses several experts where each expert learns an associated event distribution, which is then fixed 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, **Personalized 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, concordance index, and time-dependent Brier score. The contributions of our work are as follows: (i) we introduce three discrete-time deep MoE-based survival architectures, one of which achieves excellent clustering, calibration error, concordance index, and time-dependent Brier score and (ii) we report that the expressiveness of the experts is a key differentiator between discrete-time 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 patients $i = 1 \dots N$ and a the number of raw features are composed of categorical indicators (e.g., gender, etc.), and standardized continuous features (e.g., heart rate), meaning that for continuous features we subtract the mean and divide by the standard deviation. We learn embeddings for categorical indicators. After embedding, lookups are concatenated with continuous features, this 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 h-dimensional 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 head in all architectures. All methods are trained using a discrete-time Multitask Logistic Regression (MTLR) style loss function (Yu et al., 2011; Fotso, 2018) that predicts a monotone label sequence ("once event occurs, it stays on"). This loss function has been shown to be well-calibrated (Haider et al., 2020). The loss functions are described in Appendix Section A.

2.1 Fixed Mixture-of-Experts

The Fixed MoE architecture is representative of a class of prior works that use learned, but fixed per-patient, event distributions (Hou et al., 2023). The Fixed MoE 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. To our knowledge, prior discrete-time neural survival models (e.g., DeepHit (Lee et al., 2018)) do not use an explicit MoE head over a categorical time grid, and MoE-style survival models have largely been continuous-time parametric or nonparametric (Hou et al., 2023; Nagpal et al., 2021b).

2.2 Adjustable Mixture-of-Experts

The Adjustable MoE can be seen as an exemplar from a class of models that transform a prototypical event distribution per expert to form a custom event distribution per patient (Nagpal et al., 2021b;

Campanella et al., 2022; Manduchi et al., 2022). 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 $j = 0, \ldots, m-1$. 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 Section C 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., 2021b; Campanella et al., 2022; Manduchi et al., 2022).

2.3 Personalized Mixture-of-Experts

The Personalized MoE architecture belongs to a collection of models that provide the most flexibility to the experts to form custom event distributions per patient (e.g., Deep Survival Machines (Nagpal et al., 2021a)). Since this architecture generates new expert distributions for each patient, we first project the final hidden state representation $\mathbf{x}$ 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

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

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

where $M(\mathbf{x}_e)'_j$ is the j^{th} row of the parameter matrix $M(\mathbf{x}_e)$ after it has been normalized to form a discrete event distribution.

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 ↑	Brier (25th) ↓	Brier (50th) ↓	Brier (75th) ↓
Survival MNIST	CoxPH	0.030 (0.024)	79.16 (-13.65)	0.112 (0.083)	0.159 (0.125)	0.069 (0.059)
	RSF	0.057 (0.051)	90.06 (-2.76)	0.048 (0.019)	0.073 (0.038)	0.025 (0.015)
	MTLR	0.006 (0.000)	92.82 (0.00)	**0.029 (0.000)**	**0.034 (0.000)**	**0.010 (0.000)**
	Fixed MoE (ours)	0.008 (0.002)	**93.46 (0.65)**	**0.029 (0.000)**	**0.034 (0.000)**	**0.010 (-0.001)**
	Adjustable MoE (ours)	0.008 (0.002)	92.55 (-0.26)	0.030 (0.001)	0.037 (0.003)	0.011 (0.001)
	Personalized MoE (ours)	**0.005 (-0.001)**	92.61 (-0.21)	**0.029 (0.000)**	0.036 (0.002)	**0.010 (0.000)**
SUPPORT2	CoxPH	0.187 (0.130)	78.89 (-1.03)	0.212 (0.055)	0.209 (0.060)	0.236 (0.088)
	RSF	0.187 (0.129)	79.76 (-0.15)	0.207 (0.051)	0.203 (0.055)	0.232 (0.085)
	MTLR	0.057 (0.000)	79.91 (0.00)	0.156 (0.000)	0.149 (0.000)	0.148 (0.000)
	Fixed MoE (ours)	0.054 (-0.004)	79.78 (-0.13)	0.158 (0.001)	0.147 (-0.002)	0.145 (-0.003)
	Adjustable MoE (ours)	**0.048 (-0.009)**	79.83 (-0.08)	0.158 (0.002)	0.145 (-0.003)	0.143 (-0.005)
	Personalized MoE (ours)	**0.048 (-0.009)**	**80.84 (0.93)**	**0.154 (-0.002)**	**0.142 (-0.007)**	**0.138 (-0.009)**
Sepsis	CoxPH	0.635 (0.618)	73.36 (-15.00)	0.272 (0.253)	0.541 (0.508)	0.766 (0.727)
	RSF	0.604 (0.587)	82.69 (-5.67)	0.248 (0.230)	0.603 (0.570)	0.811 (0.773)
	MTLR	0.017 (0.000)	88.36 (0.00)	0.019 (0.000)	0.033 (0.000)	0.039 (0.000)
	Fixed MoE (ours)	0.011 (-0.006)	87.09 (-1.27)	0.019 (0.000)	0.033 (-0.000)	0.039 (0.001)
	Adjustable MoE (ours)	0.009 (-0.008)	88.99 (0.63)	**0.017 (-0.001)**	0.032 (-0.001)	0.037 (-0.002)
	Personalized MoE (ours)	**0.005 (-0.012)**	**89.77 (1.41)**	**0.017 (-0.002)**	**0.030 (-0.003)**	**0.036 (-0.003)**

3 Experiments

We experiment with 3 datasets to probe the properties and capabilities of our proposed methods. Survival MNIST 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 were summarized to perform a retrospective prediction, helpful when using partially labeled historical datasets to accelerate human labeling. Patients in the Sepsis dataset were administratively censored after 100 hours. As a comprehensive set of metrics covering performance with respect to accuracy and uncertainty calibration, we measure Harrell's concordance index (Harrell et al., 1996), the time-dependent Brier score (standard inverse probability of censoring weighting (IPCW) (Gerds & Schumacher, 2006)) at 3 key points in time: the 25th, 50th, and 75th percentile of time bins, and equal mass expected calibration error (ECE) (Roelofs et al., 2022) adjusted with IPCW and averaged over all time bins. 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 Appendix Section B.

4 Results

We report our results in Table 1. The Survival MNIST dataset represents the Platonic ideal of a dataset with clear latent groups and as a result the Fixed MoE model performs best across all metrics except for calibration. This is a setting where the Fixed 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 can hurt performance. Nevertheless, the Personalized MoE model is best calibrated and matches the performance of the Fixed MoE on Brier at the 25th and 75th percentiles. The Survival MNIST dataset provides an interesting contrast to real-world datasets, where latent groups are almost never as well-defined.

Results on SUPPORT2 show the Personalized MoE model outperforms all other methods across all metrics. Importantly, the Personalized MoE model is outperforming the MTLR model on calibration error, concordance, and Brier score at all time points, while the fixed and adjustable MoE models are, in general, not. On SUPPORT2, our concordance and IPCW Brier scores are in line with prior reports for modern deep survival models (e.g., DSM (Nagpal et al., 2021a)). We see similarly excellent results on the Sepsis dataset, providing further evidence that the Personalized MoE model is able

to deliver on all desiderata: clustering, calibration, and predictive accuracy. Consistent with prior work (Nagpal et al., 2021b; Manduchi et al., 2022), we observe that per-patient adjustments can benefit calibration and predictive accuracy as measured by the Brier score relative to MTLR. Our Personalized MoE model extends those gains, delivering simultaneous improvements in calibration, accuracy, and discrimination, achieved with simple end-to-end learning rather than more complex training pipelines (e.g., spline estimation of baseline hazard rates (Nagpal et al., 2021b), EM-based cluster estimation (Buginga & e Silva, 2024), etc.). The Personalized 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 Survival MNIST.

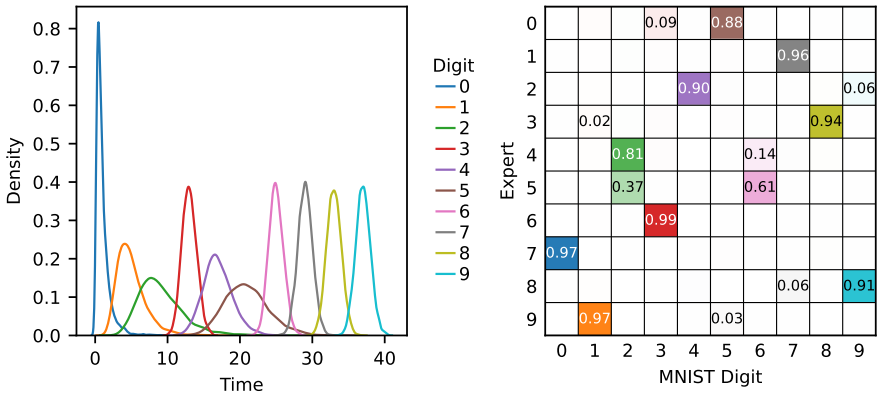

Figure 1: The left plot shows synthetic event distributions for each MNIST digit in Survival MNIST and the right plot shows which digits get routed to each expert in a **Personalized MoE** model.

We report routing analysis results in Figure 1, where each row of the matrices represents the distribution over Survival MNIST digits routed to that particular expert. For example, in Figure 1, among all data points that are routed to expert 0 in a Personalized MoE, 88% are digit 5. Similarly, for expert 1, 96% are digit 7, and so on. For Figure 1, the routing decision for a data instance is made by selecting the expert with the highest weight. This degree of specialization per expert indicates that these models are able to recover latent groups in the Survival MNIST dataset.

5 Conclusion

We developed three novel deep mixture-of-experts (MoE)-based survival architectures, one of which achieves excellent clustering, calibration error, and predictive accuracy. 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 MoEs given the rich space of possible parameter-efficient transformations of expert distributions, and how patients are routed to experts on real-world datasets.

Acknowledgments and Disclosure of Funding

We would like to thank Jake Snell, Thomas Zollo, Zhun Deng, Kayla Schiffer-Kane, and Emily Saunders for their helpful discussions. This publication was supported by the National Center for Advancing Translational Sciences, National Institutes of Health, through Grant Number UL1TR001873. The content is solely the responsibility of the authors and does not necessarily represent the official views of the NIH. The authors also acknowledge support from the National Science Foundation and by DoD OUSD (R&E) under Cooperative Agreement PHY-2229929 (The NSF AI Institute for Artificial and Natural Intelligence) (RZ). This work was also supported in part by National Institutes of Health: 1R01NS129760-01 (SP), 1R01NS131606-01 (SP), American Heart Association: 24SCEFIA1259295 (MM).

References

Ahsan, M. M., Luna, S. A., and Siddique, Z. Machine-Learning-Based Disease Diagnosis: A Comprehensive Review. *Healthcare*, 10(3):541, March 2022. ISSN 2227-9032. doi: 10.3390/healthcare10030541.

Buginga, G. and e Silva, E. d. S. Clustering Survival Data using a Mixture of Non-parametric Experts, May 2024.

Campanella, G., Kook, L., Häggström, I., Hothorn, T., and Fuchs, T. J. Deep conditional transformation models for survival analysis, October 2022.

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.

Erickson, B. J., Korfiatis, P., Akkus, Z., and Kline, T. L. Machine Learning for Medical Imaging. *RadioGraphics*, 37(2):505–515, March 2017. ISSN 0271-5333. doi: 10.1148/rg.2017160130.

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

Gerds, T. A. and Schumacher, M. Consistent estimation of the expected Brier score in general survival models with right-censored event times. *Biometrical Journal. Biometrische Zeitschrift*, 48 (6):1029–1040, December 2006. ISSN 0323-3847. doi: 10.1002/bimj.200610301.

Haider, H., Hoehn, B., Davis, S., and Greiner, R. Effective ways to build and evaluate individual survival distributions. *J. Mach. Learn. Res.*, 21(1):85:3289–85:3351, January 2020. ISSN 1532-4435.

Harrell, F. E., Lee, K. L., and Mark, D. B. Multivariable prognostic models: Issues in developing models, evaluating assumptions and adequacy, and measuring and reducing errors. *Statistics in Medicine*, 15(4):361–387, February 1996. ISSN 0277-6715. doi: 10.1002/(SICI)1097-0258(19960229)15:4<361::AID-SIM168>3.0.CO;2-4.

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.

Lee, C., Zame, W., Yoon, J., and van der Schaar, M. DeepHit: A Deep Learning Approach to Survival Analysis With Competing Risks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), April 2018. ISSN 2374-3468. doi: 10.1609/aaai.v32i1.11842.

Manduchi, L., Marcinkevičs, R., Massi, M. C., Weikert, T., Sauter, A., Gotta, V., Müller, T., Vasella, F., Neidert, M. C., Pfister, M., Stieltjes, B., and Vogt, J. E. A Deep Variational Approach to Clustering Survival Data. In *International Conference on Learning Representations*, March 2022. doi: 10.48550/arXiv.2106.05763.

Nagpal, C., Li, X., and Dubrawski, A. Deep Survival Machines: Fully Parametric Survival Regression and Representation Learning for Censored Data With Competing Risks. *IEEE Journal of Biomedical and Health Informatics*, 25(8):3163–3175, August 2021a. ISSN 2168-2208. doi: 10.1109/JBHI.2021.3052441.

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 2021b.

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.

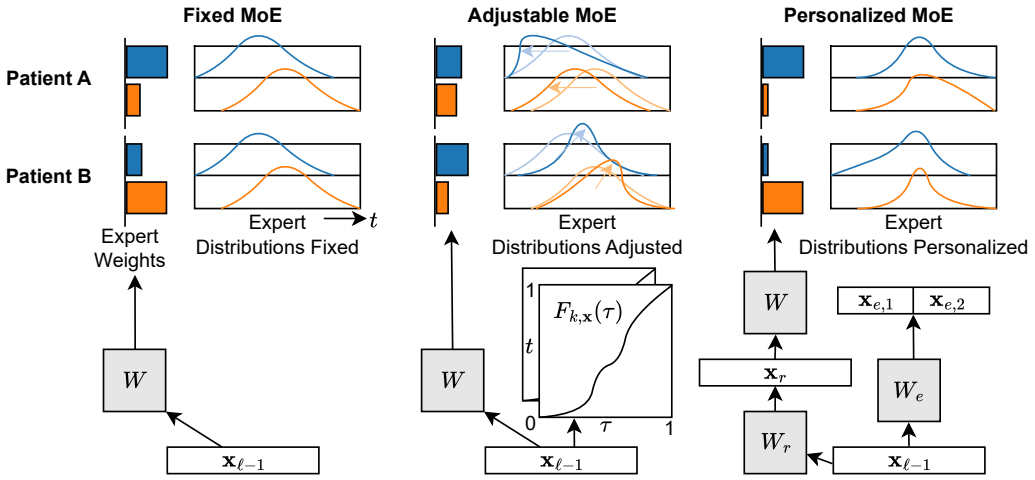

Figure 2: Illustration of our three proposed mixture-of-experts (MoE) architectures for survival analysis. Left - **Fixed MoE** showing the same expert distributions for patients A and B with differing expert weights. Middle - **Adjustable MoE** where fixed expert distributions (faint color) are adjusted per patient (dark color). Right - **Personalized MoE** where all experts produce a custom event distribution for all patients.

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 softmax($\mathbf{u}$) for $\mathbf{u} \in \mathbb{R}^m$. Recall that the softmax function is invariant to additive shifts (i.e., softmax($\mathbf{u}$) = 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

We define a validation set for Survival MNIST 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 loss to perform hyperparameter tuning for each dataset and method. The temperature parameter κ is initialized to 2.0 for all MoE models and learned during training. The load balancing loss λ_{lb} is constrained. For instance, setting the load balancing loss to be 0 or too close to 0 results in the model not using all of the experts. We found $\lambda_{\text{lb}} = 0.01$ to consistently allow models to use most, if not all, the experts, while still allowing for specialization. We tuned the learning rate for all neural methods from the set {5e-3, 5e-4, 5e-5}. All models are trained with the Adam optimizer. For the Cox Proportional Hazards model we tuned the `alphas` hyperparameter from the set $\{0.001, 0.01, 0.1\}$ and the `l1_ratio` from the set $\{0.01, 0.5, 1.0\}$. The Random Survival Forest model's `n_estimators` hyperparameter was tuned from the set $\{50, 100, 200\}$ and the `min_samples_split` hyperparameter was tuned from the set $\{50, 100, 200\}$. Due to long runtimes and high memory requirements on the

Table 2: Parameter counts for all neural methods.

Dataset	Model	Parameters
Survival MNIST	MTLR	187,189
	Fixed MoE	209,844
	Adjustable MoE	194,883
	Personalized MoE	195,891
SUPPORT2	MTLR	68,521
	Fixed MoE	69,480
	Adjustable MoE	69,435
	Personalized MoE	62,141
Sepsis	MTLR	62,945
	Fixed MoE	63,008
	Adjustable MoE	63,579
	Personalized MoE	57,909

Table 3: Model specific hyperparameters for Survival MNIST.

Model	Fixed MoE	Adjustable MoE	Personalized MoE	MTLR
Hidden Dim. (h)	208	186	160	176
Num. Layers (ℓ)	2	2	1	2
Num. Experts (n)	10	10	10	-
Learning Rate	5e-4	5e-4	5e-4	5e-4

Table 4: Model specific hyperparameters for SUPPORT2 dataset.

Model	Fixed MoE	Adjustable MoE	Personalized MoE	MTLR
Hidden Dim. (h)	176	186	128	176
Num. Layers (ℓ)	2	2	1	2
Num. Experts (n)	10	10	8	-
Learning Rate	5e-3	5e-3	5e-4	5e-4

Table 5: Model specific hyperparameters for Sepsis dataset.

Model	Fixed MoE	Adjustable MoE	Personalized MoE	MTLR
Hidden Dim. (h)	176	186	128	176
Num. Layers (ℓ)	2	2	1	2
Num. Experts (n)	10	10	8	-
Learning Rate	5e-4	5e-4	5e-4	5e-4

Table 6: Shared hyperparameters for all models.

Hyperparameter	Value
Batch Size	64
λ_{lb}	0.01
κ Init.	2.0
Time Bins (m)	100

Survival MNIST dataset–likely due to the large number of features–we directly set `n_estimators` and `min_samples_split` to 100 and 100 respectively without tuning. We set hidden dimension sizes so that parameter counts are approximately equal across neural methods to ensure a fair comparison

Table 7: Random survival forest hyperparameters using the `scikit-survival` library.

Hyperparameter	Survival MNIST	Sepsis	SUPPORT2
`n_estimators`	100	50	200
`max_features`	`sqrt`	`sqrt`	`sqrt`
`min_samples_split`	100	50	200

Table 8: Cox proportional hazards hyperparameters for all datasets using the `scikit-survival` library.

Hyperparameter	Value
`fit_baseline`	`True`
`alphas`	`[0.01]`
`l1_ratio`	`0.01`

and isolate the effects of the decoder head used. We use a hidden dimension of 128-208 and 1-2 hidden layers for all models, which are fully connected layers followed by 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. 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 implemented in PyTorch and trained on a single GPU. Our GitHub code repository is available at `https://github.com/ToddMorrill/survival-moe`. We report the final hyperparameters used for each model and dataset in the tables above.

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 $\text{lo} \leftarrow \mathbf{1}_{\{v<0\}} \cdot \text{mid} + \mathbf{1}_{\{v\geq 0\}} \cdot \text{lo}$, $\text{hi} \leftarrow \mathbf{1}_{\{v<0\}} \cdot \text{hi} + \mathbf{1}_{\{v\geq 0\}} \cdot \text{mid}$.

3. Return $\tau^\star \approx (\text{lo} + \text{hi})/2$.

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 $\partial \tau^\star / \partial \theta$.