High-Cardinality Categorical Covariates in Pricing

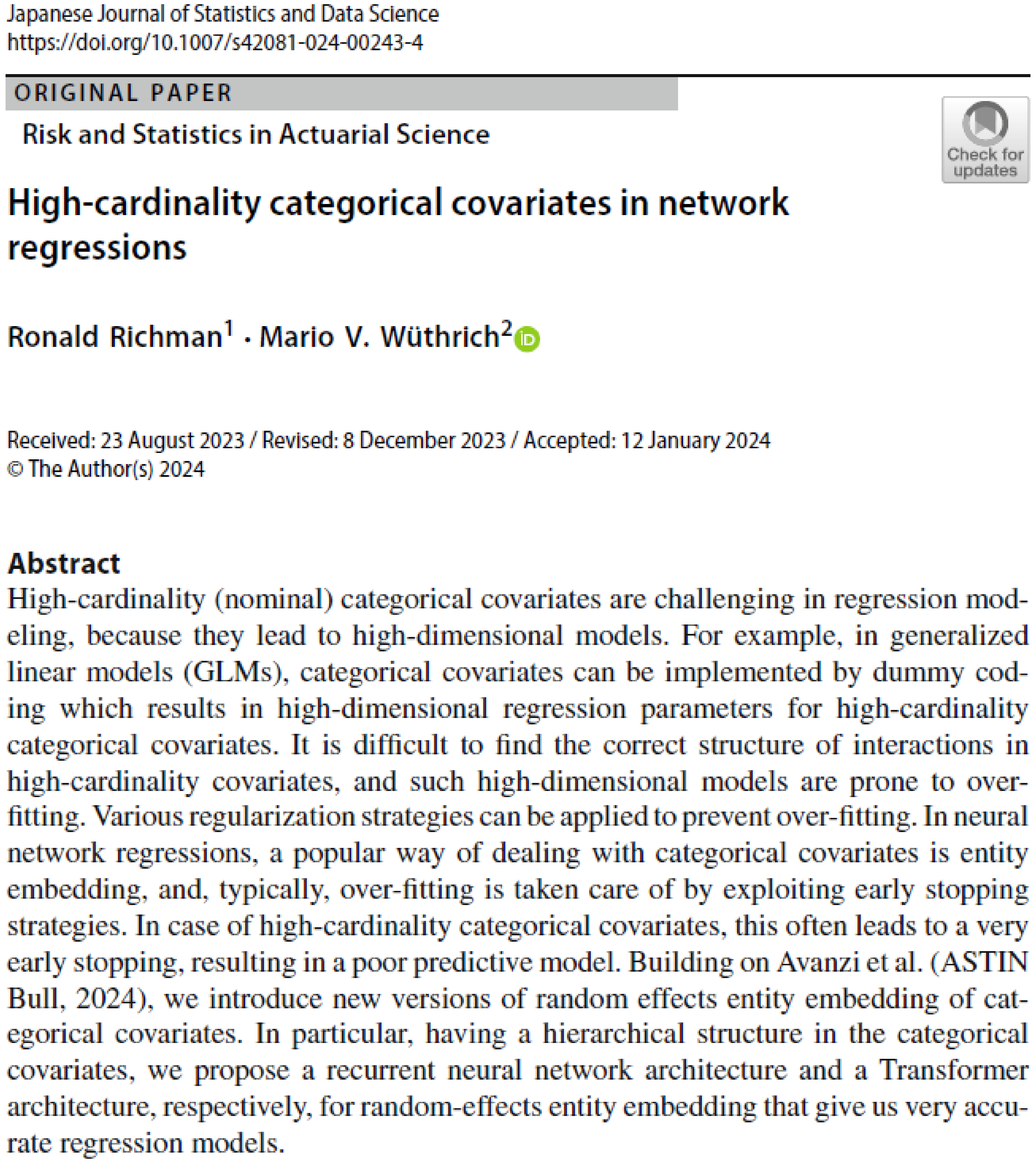
Mario Wüthrich – ETH Zurich

IAK The Institute of Actuaries of Korea

11 September 2024

Background

- **Joint work with Ronald Richman (Old Mutual Insure)**
- **Paper: High-cardinality categorical covariates in network regressions**
- **Available for free from: https://link.springer.com/article/10.1007/s42081-024-00243-4**
- **Code: https://github.com/wueth/High-Cardinality-Covariates-Regularization**



Japanese Journal of Statistics and Data Science
https://doi.org/10.1007/s42081-024-00243-4

ORIGINAL PAPER

Risk and Statistics in Actuarial Science

High-cardinality categorical covariates in network regressions

Ronald Richman[1] · Mario V. Wüthrich[2]

Received: 23 August 2023 / Revised: 8 December 2023 / Accepted: 12 January 2024

Abstract

High-cardinality (nominal) categorical covariates are challenging in regression modeling, because they lead to high-dimensional models. For example, in generalized linear models (GLMs), categorical covariates can be implemented by dummy coding which results in high-dimensional regression parameters for high-cardinality categorical covariates. It is difficult to find the correct structure of interactions in high-cardinality covariates, and such high-dimensional models are prone to over-fitting. Various regularization strategies can be applied to prevent over-fitting. In neural network regressions, a popular way of dealing with categorical covariates is entity embedding, and, typically, over-fitting is taken care of by exploiting early stopping strategies. In case of high-cardinality categorical covariates, this often leads to a very early stopping, resulting in a poor predictive model. Building on Avanzi et al. (ASTIN Bull, 2024), we introduce new versions of random effects entity embedding of categorical covariates. In particular, having a hierarchical structure in the categorical covariates, we propose a recurrent neural network architecture and a Transformer architecture, respectively, for random-effects entity embedding that give us very accurate regression models.

Outlook for the talk

- **Introduce methods for dealing with categorical covariates…**
- **… with high cardinality = many levels**
- **Discuss regularization to ensure the estimates have credibility**
- **Discuss how to deal with categorical covariates with a natural hierarchy**
- **Tools used: embeddings and deep learning**
- **Show results on a simulated dataset**

Agenda

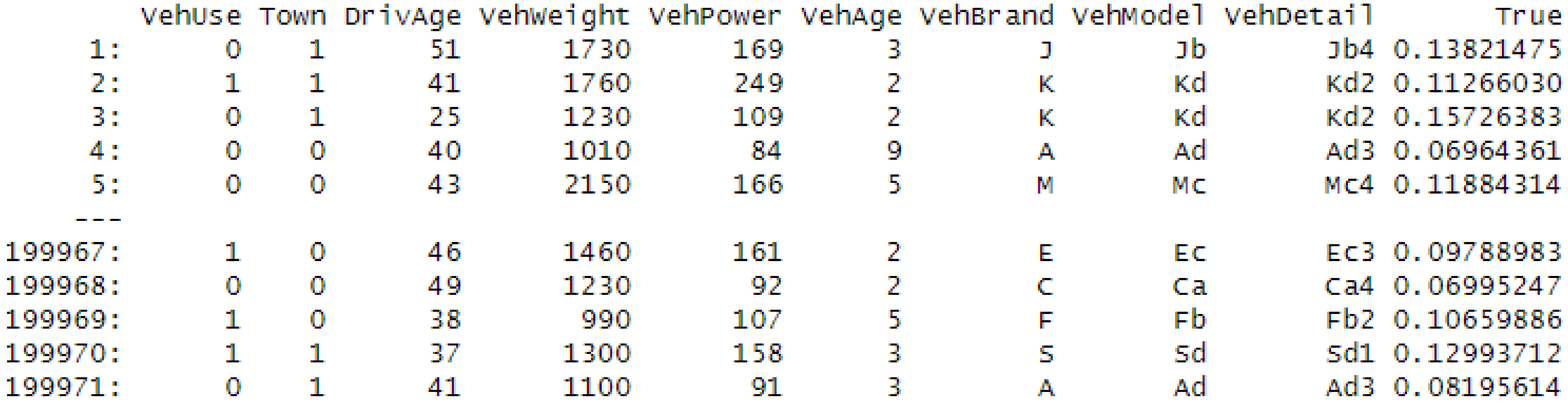
- <u>**What are embeddings?**</u>
- **GLMMs**
- **Let’s think Bayesian**
- **Hierarchical models based on embeddings**

Pricing with categorical covariates

- **Most insurance pricing datasets have a large number of categorical covariates in addition to continuous covariates**

- **Categorical covariates may also arise from pre-processing continuous covariates into categorical ones (binning), e.g., age classes or vehicle power classes**

- **Simulated dataset used throughout as example**



```
        VehUse Town DrivAge VehWeight VehPower VehAge VehBrand VehModel VehDetail       True
     1:      0    1      51      1730      169      3        J       Jb       Jb4 0.13821475
     2:      1    1      41      1760      249      2        K       Kd       Kd2 0.11266030
     3:      0    1      25      1230      109      2        K       Kd       Kd2 0.15726383
     4:      0    0      40      1010       84      9        A       Ad       Ad3 0.06964361
     5:      0    0      43      2150      166      5        M       Mc       Mc4 0.11884314
    ---
199967:      1    0      46      1460      161      2        E       Ec       Ec3 0.09788983
199968:      0    0      49      1230       92      2        C       Ca       Ca4 0.06995247
199969:      1    0      38       990      107      5        F       Fb       Fb2 0.10659886
199970:      1    1      37      1300      158      3        S       Sd       Sd1 0.12993712
199971:      0    1      41      1100       91      3        A       Ad       Ad3 0.08195614
```

GLMs with categorical covariates

- **Actuarial pricing often uses GLMs which has a linear model form after applying the link function:**

$$\boldsymbol{g^{-1}(\hat{y}) = \beta_0 + \beta_1 . x_1 + \cdots + \beta_q . x_q}$$

- **Suppose $\mathbf{x_1}$ is categorical (or continuous and was binned), then we can represent it in the GLM as**

$$\boldsymbol{g^{-1}(\hat{y}) = \begin{cases} \beta_0 + \quad 0 \quad + \cdots + \beta_q . x_q & when\, x_1 = x_{1,1} \\ \beta_0 + \beta_{1,2} + \cdots + \beta_q . x_q & when\, x_1 = x_{1,2} \\ & \\ \beta_0 + \beta_{1,L} + \cdots + \beta_q . x_q & when\, x_1 = x_{1,L} \end{cases}}$$

- **In a GLM, as we vary $\boldsymbol{x_{1,1} \rightarrow x_{1,2}}$ our GLM value changes from 0 (reference level) to $\boldsymbol{\beta_{1,2}}$ (uses dummy coding)**

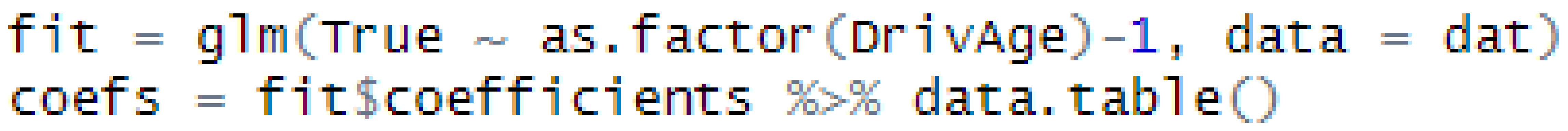
- **$\boldsymbol{\beta}$ coefficients are estimated directly from the data…**

- **… in deep learning, these coefficients called embeddings**

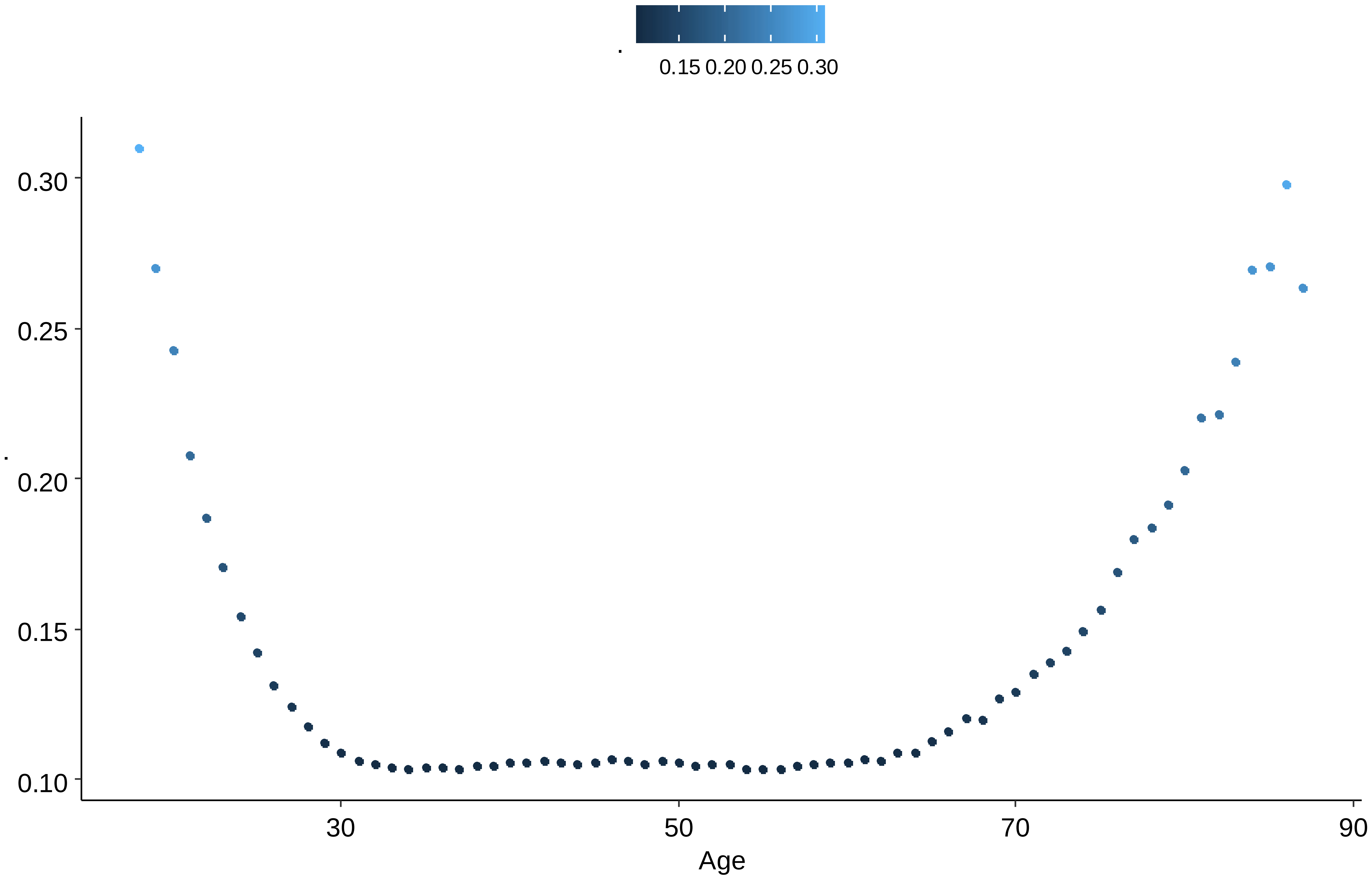
- **This is a 1-dimensional embedding while we can generalize things to multiple dimensions**

GLM example

- **Simple GLM regressing TRUE frequency from driver age…**
- **… produces GLM coefficients which are:**
  - Pricing relativities
  - 1D embeddings
- **Since output is frequency, coefficients can be interpreted on frequency scale**
- **In this case, DrivAge is treated as categorical whereas one could also treated it as continuous =>**
- **In that case it has only a single β coefficient**
- **We present deep learning models and higher dimensional embeddings**



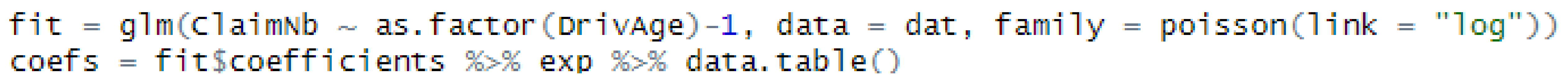
```
fit = glm(True ~ as.factor(DrivAge)-1, data = dat)
coefs = fit$coefficients %>% data.table()
```



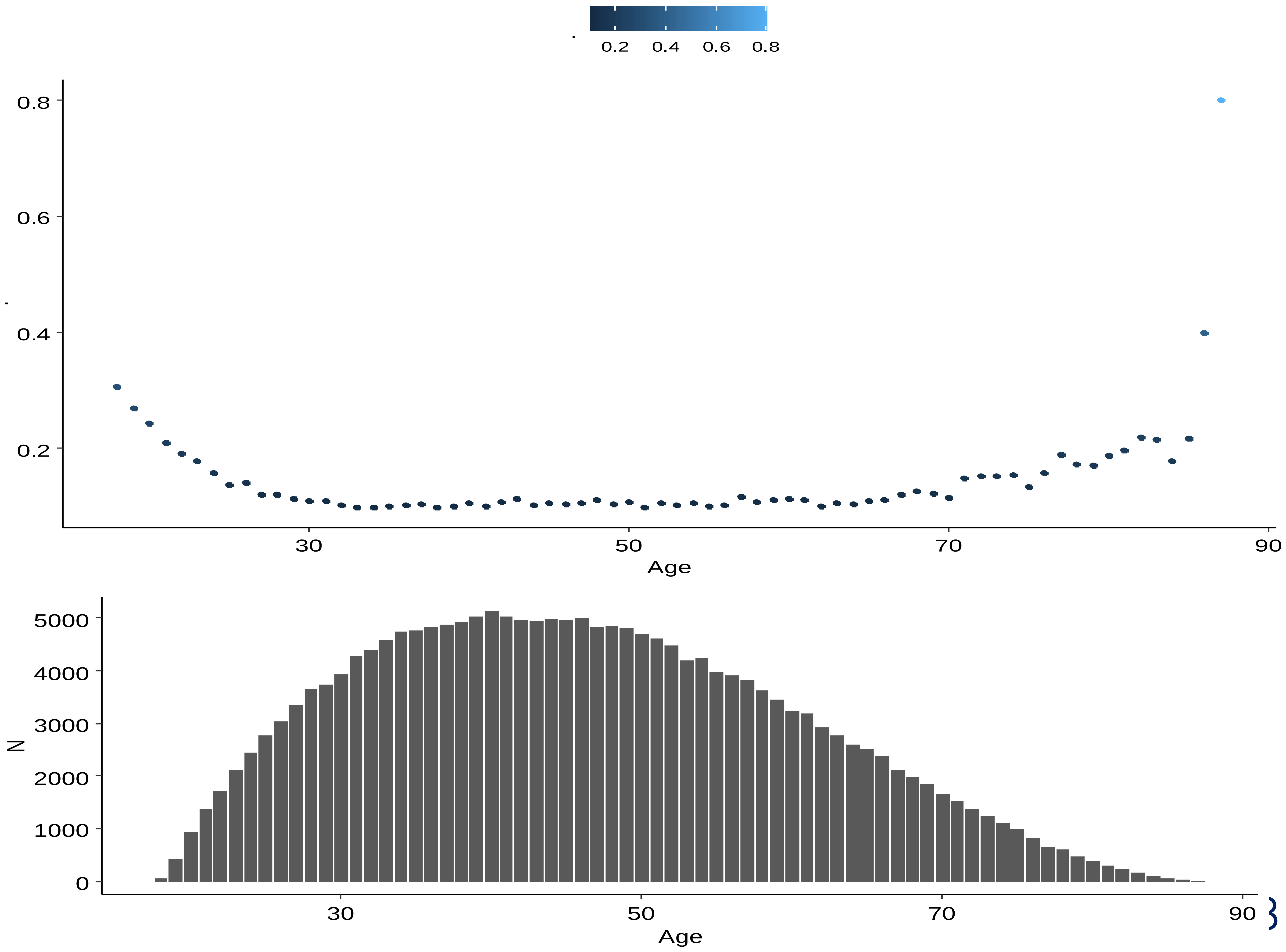
GLM example (2)

- **We do not know the TRUE frequency, so need to estimate from (very) noisy experience data**
- **Example – where we have little exposure, we struggle to get accurate results of the TRUE frequency**
- **=> (1) We may want to smooth rates out,**
- **Or (2) apply regularization – main theme of the paper,**
- **Or (3) can smooth out by hand but:**

  Highly manual and time consuming
  Depends on expert judgement



```
fit = glm(ClaimNb ~ as.factor(DrivAge)-1, data = dat, family = poisson(link = "log"))
coefs = fit$coefficients %>% exp %>% data.table()
```

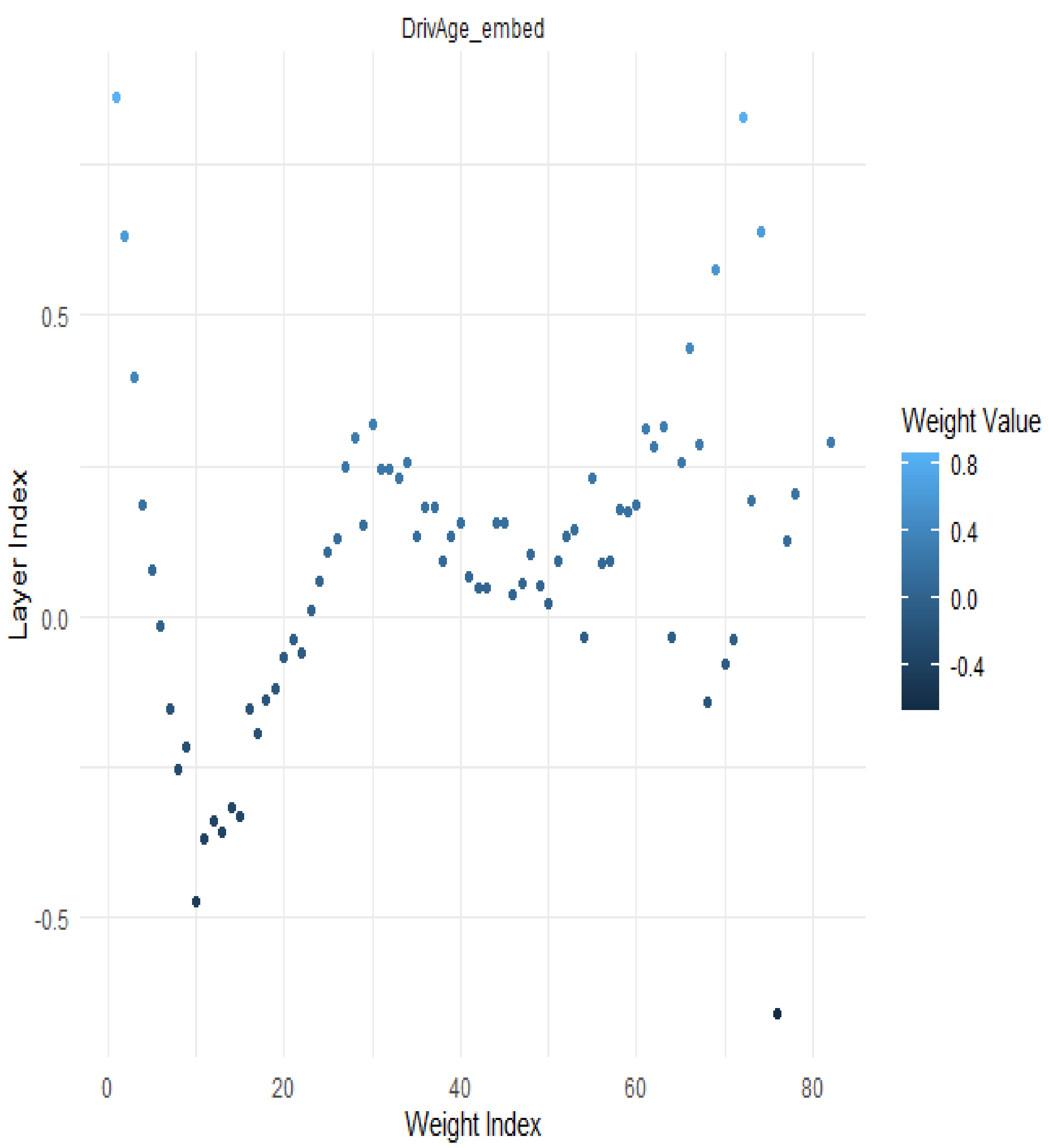


Two types of embeddings

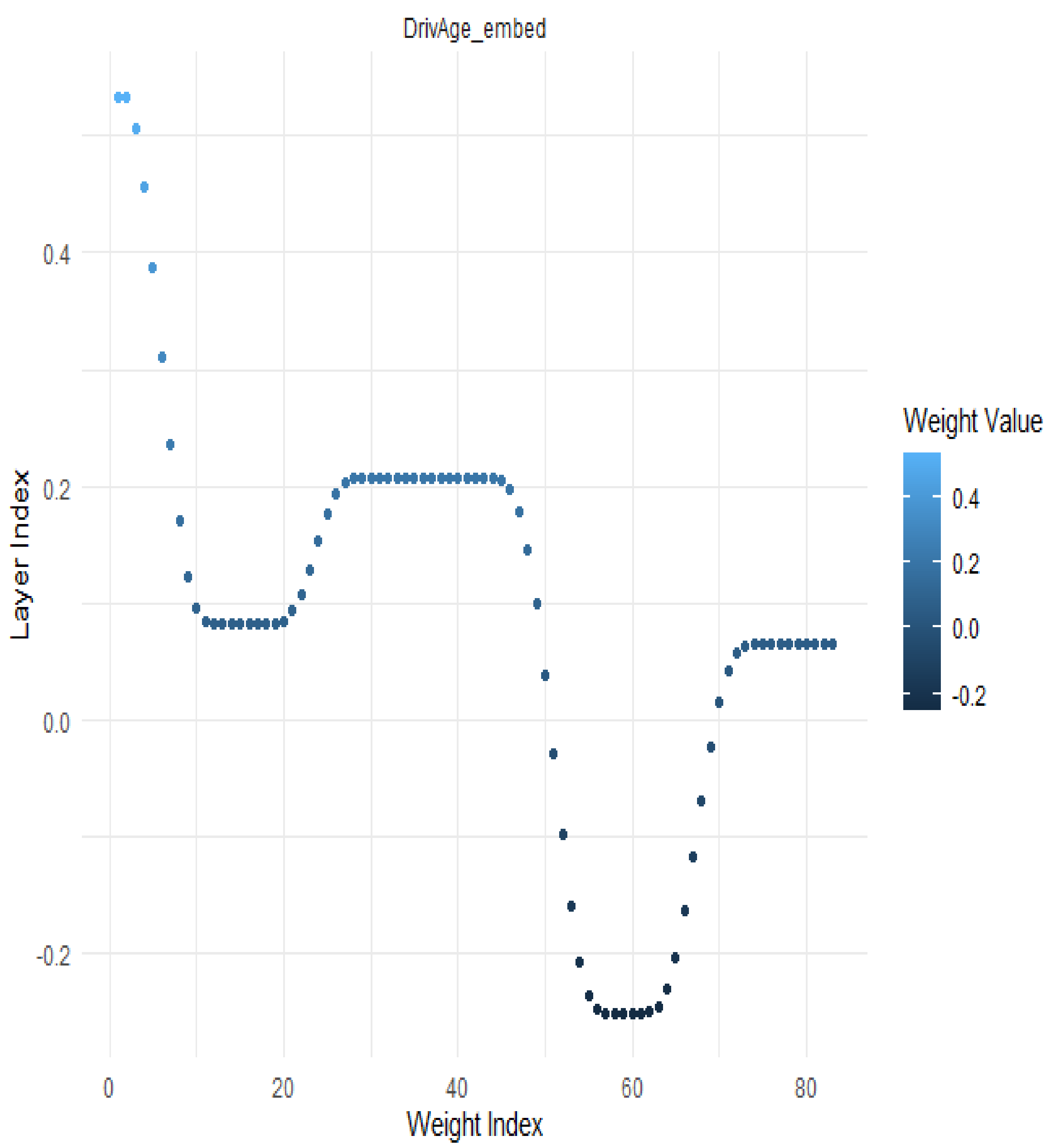
- **Continuous data that has been binned has a natural order (ordinal) – so one can smooth it by applying Fused LASSO regularization, see example below.**

- **Categorical data does not have a natural order (nominal) – so one needs to think about other approaches of regularization, usually similar to credibility theory in some way.**

Raw

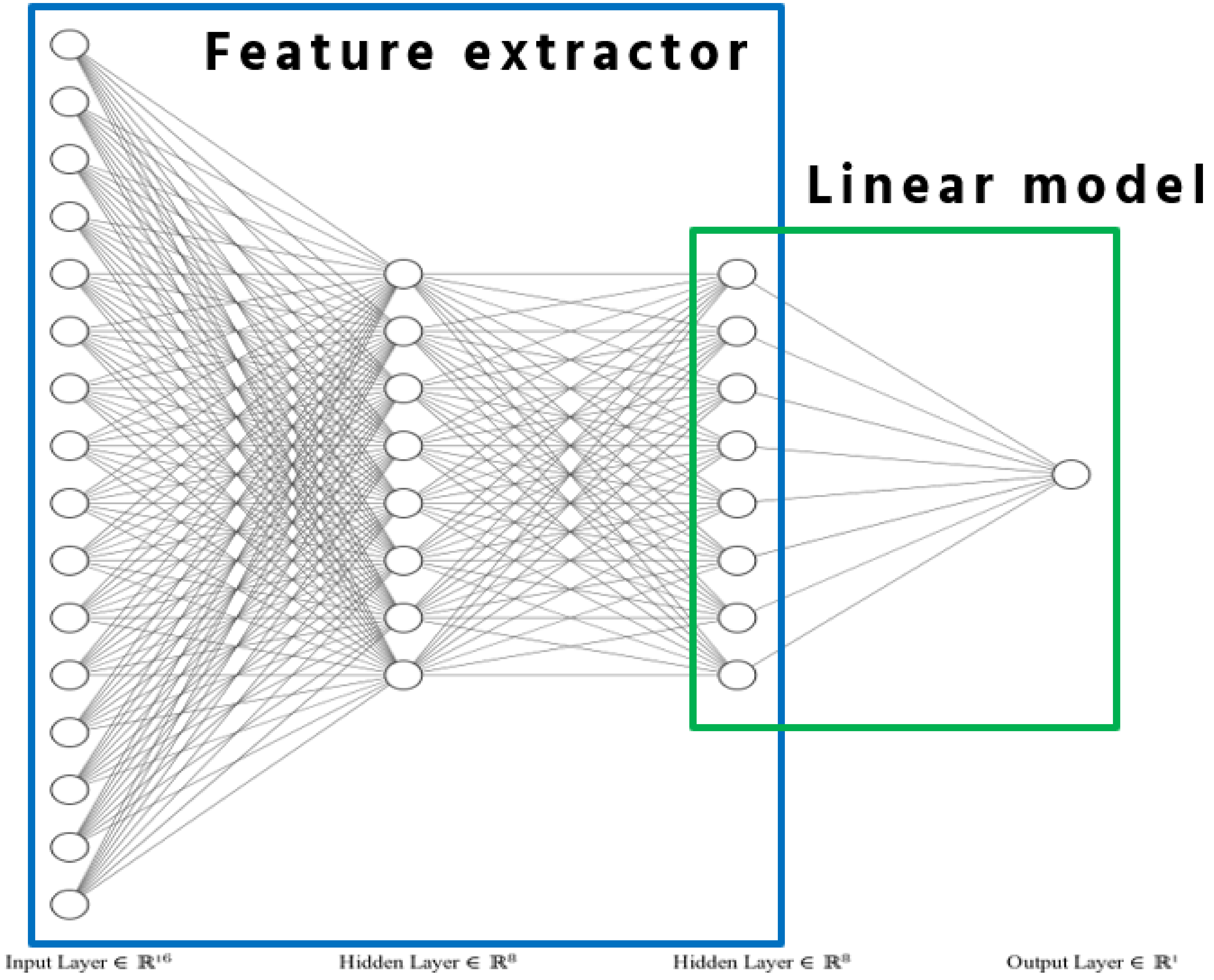


Smoothed



Deep learning - FCN generalizes GLM

- A fully connected network (FCN) generalizes a GLM
- Intermediate layers = representation learning, guided by supervised objective
- Last layer = generalized linear model (GLM), where input variables = new representation of data (from feature extractor)
- No need to use GLM – strip off last layer and use learned features in, for example, XGBoost regression


Feature extractor
Linear model
Input Layer $\in \mathbb{R}^{16}$
Hidden Layer $\in \mathbb{R}^{8}$
Hidden Layer $\in \mathbb{R}^{8}$
Output Layer $\in \mathbb{R}^{1}$

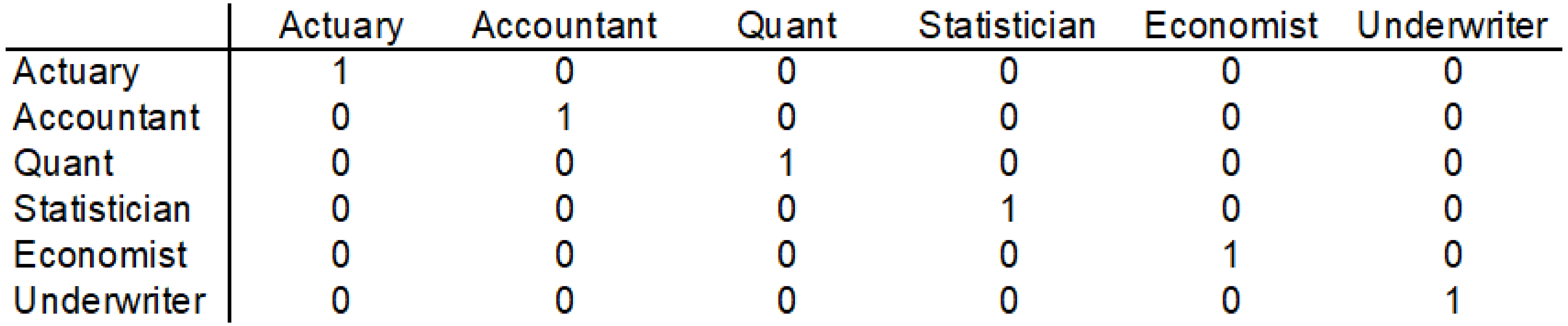
Embedding layer – categorical data

- **One-hot encoding maps each categorical level to a basis vector (0,...,0,1,0,...,0)**

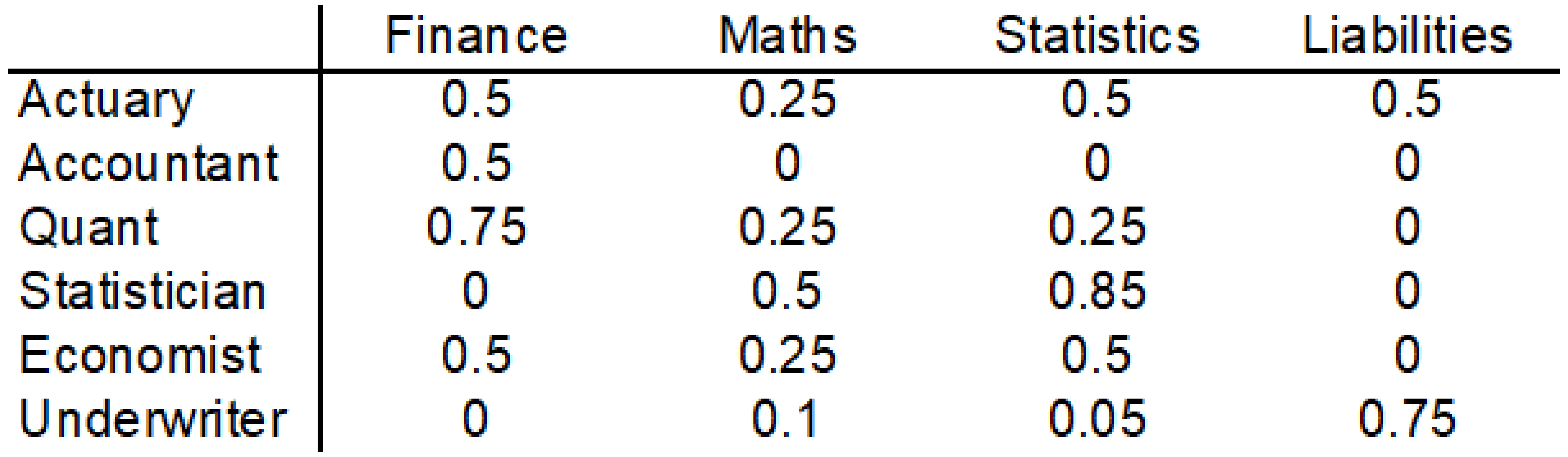
- **One-hot encoding expresses a prior model with categories being orthogonal => similar levels are not categorized into groups**

- **Embedding layer – similar categories should cluster together:**

  Learn dense vector transformation of sparse input vectors and cluster similar categories together



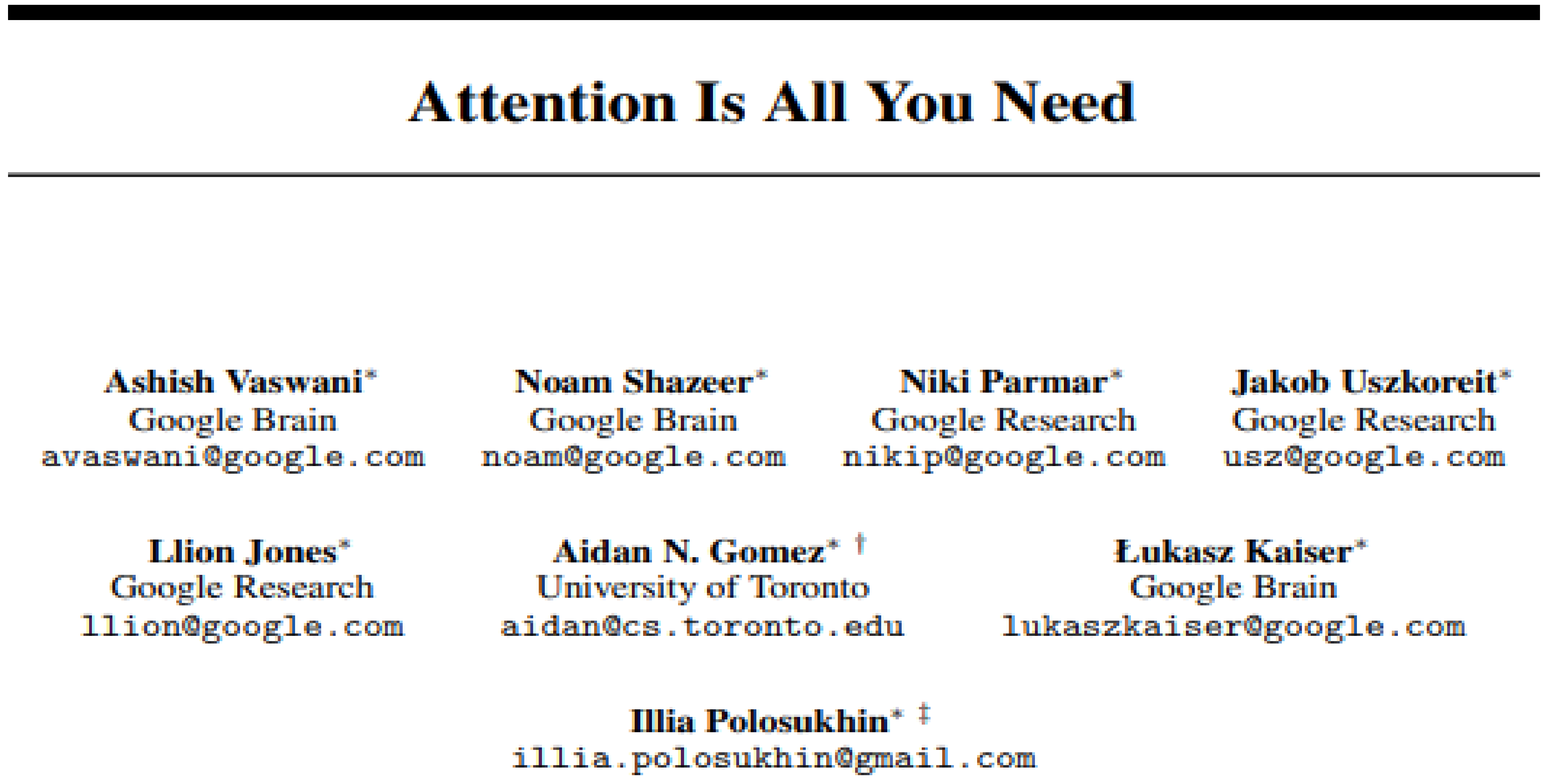
	Actuary	Accountant	Quant	Statistician	Economist	Underwriter
Actuary	1	0	0	0	0	0
Accountant	0	1	0	0	0	0
Quant	0	0	1	0	0	0
Statistician	0	0	0	1	0	0
Economist	0	0	0	0	1	0
Underwriter	0	0	0	0	0	1



	Finance	Maths	Statistics	Liabilities
Actuary	0.5	0.25	0.5	0.5
Accountant	0.5	0	0	0
Quant	0.75	0.25	0.25	0
Statistician	0	0.5	0.85	0
Economist	0.5	0.25	0.5	0
Underwriter	0	0.1	0.05	0.75

Modern ML relies on embeddings!

- ML approaches rely on learning embeddings for natural language processing (NLP) and applying Transformers to these (ChatGPT)
- Approach proposed in 2017 relies on attention mechanisms
- Extended to other tasks such as computer vision and more recently, tabular data, Kuo-Richman.



Attention Is All You Need

Ashish Vaswani[*]
Google Brain
avaswani@google.com

Noam Shazeer[*]
Google Brain
noam@google.com

Niki Parmar[*]
Google Research
nikip@google.com

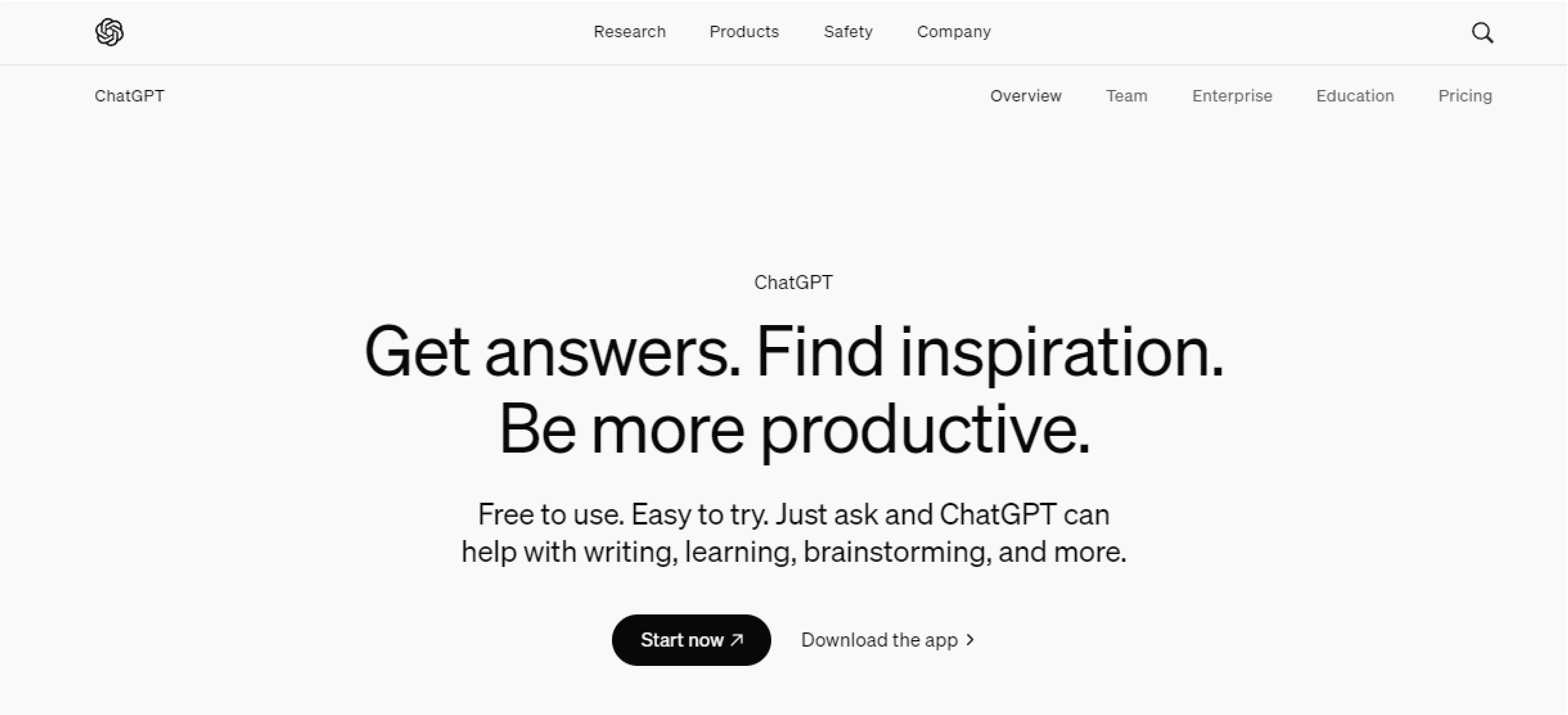
Jakob Uszkoreit[*]
Google Research
usz@google.com

Llion Jones[*]
Google Research
llion@google.com

Aidan N. Gomez[*] [†]
University of Toronto
aidan@cs.toronto.edu

Łukasz Kaiser[*]
Google Brain
lukaszkaiser@google.com

Illia Polosukhin[*] [‡]
illia.polosukhin@gmail.com



ChatGPT

Get answers. Find inspiration.
Be more productive.

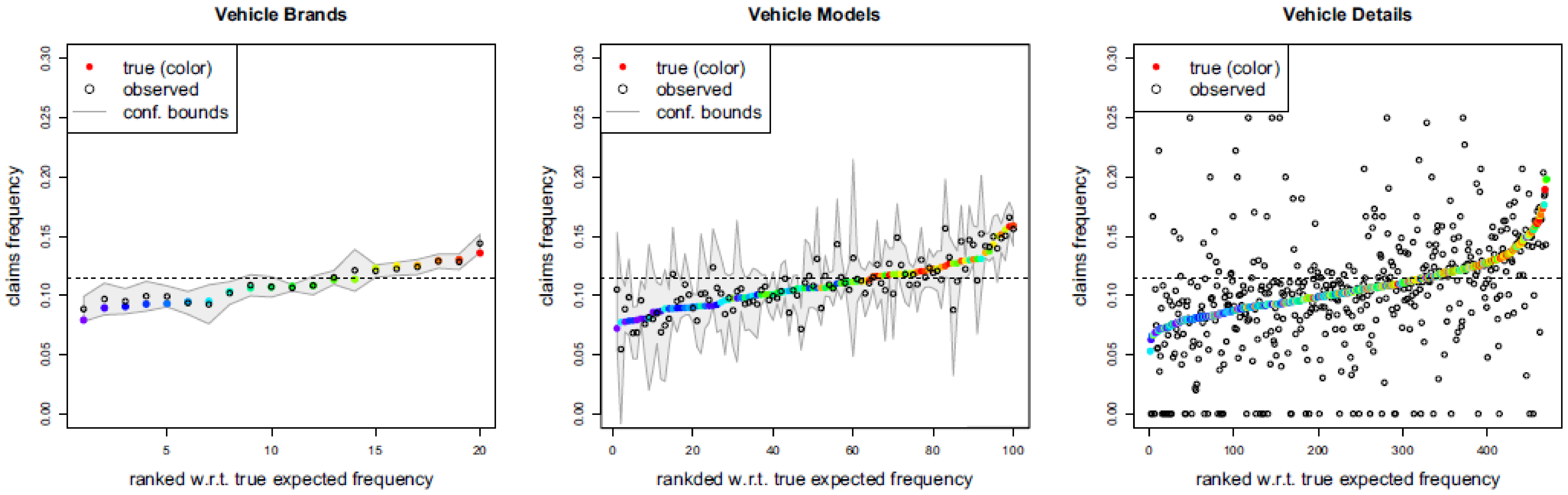
Free to use. Easy to try. Just ask and ChatGPT can help with writing, learning, brainstorming, and more.

Agenda

- **What are embeddings?**
- **GLMMs**
- **Let’s think Bayesian**
- **Hierarchical models based on embeddings**

What can go wrong with embeddings?

- **We may have categorical variables where there are very few examples of each level**
- **E.g.: Car vehicle brand/model/variant – lots of Kias but very few Abarths!**
- **Noisy data generated by claims process =>**
- **May not have enough credibility to set values of factor levels/embeddings appropriately**
- **Example from simulated data (colour = Vehicle Brand)**

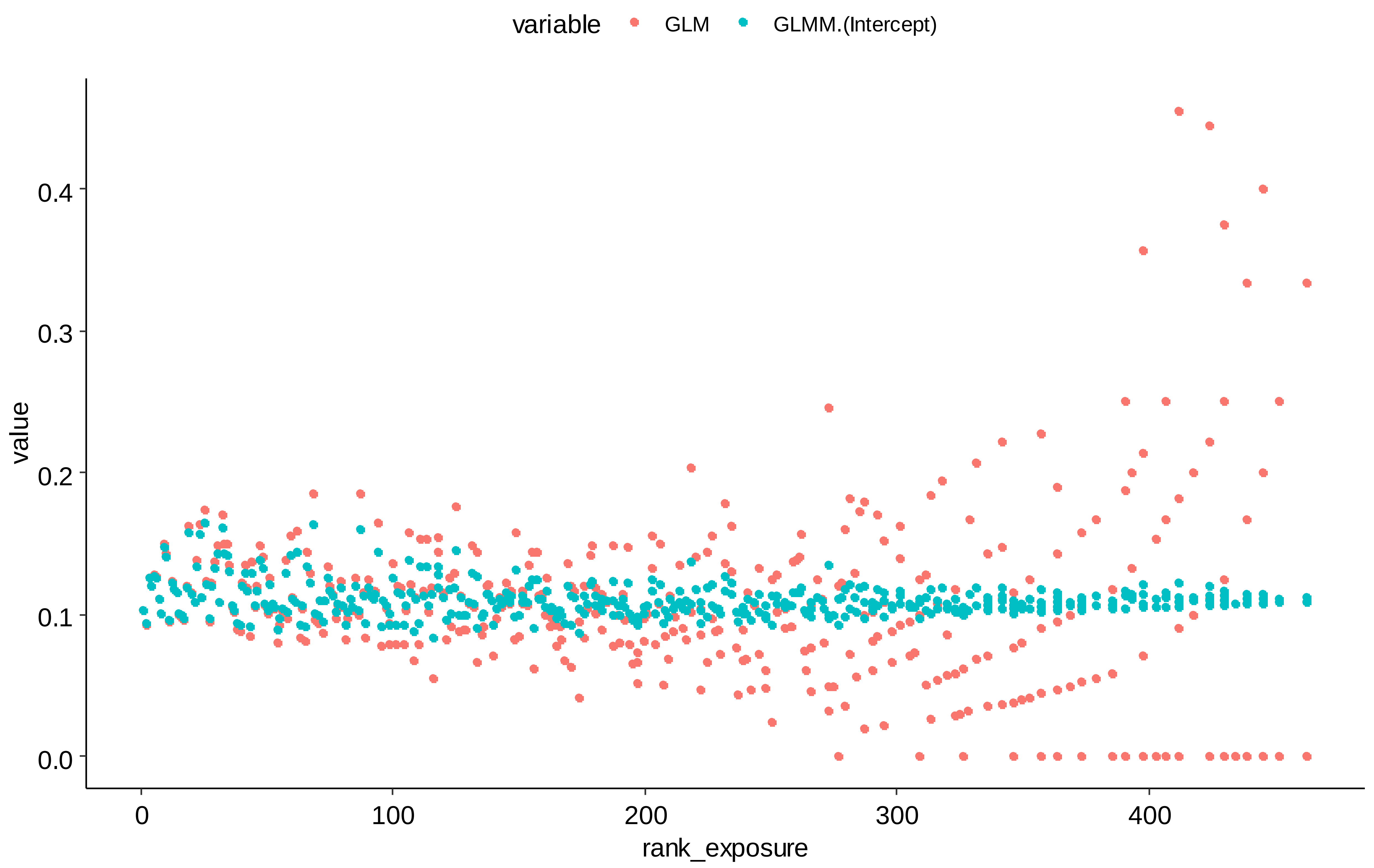


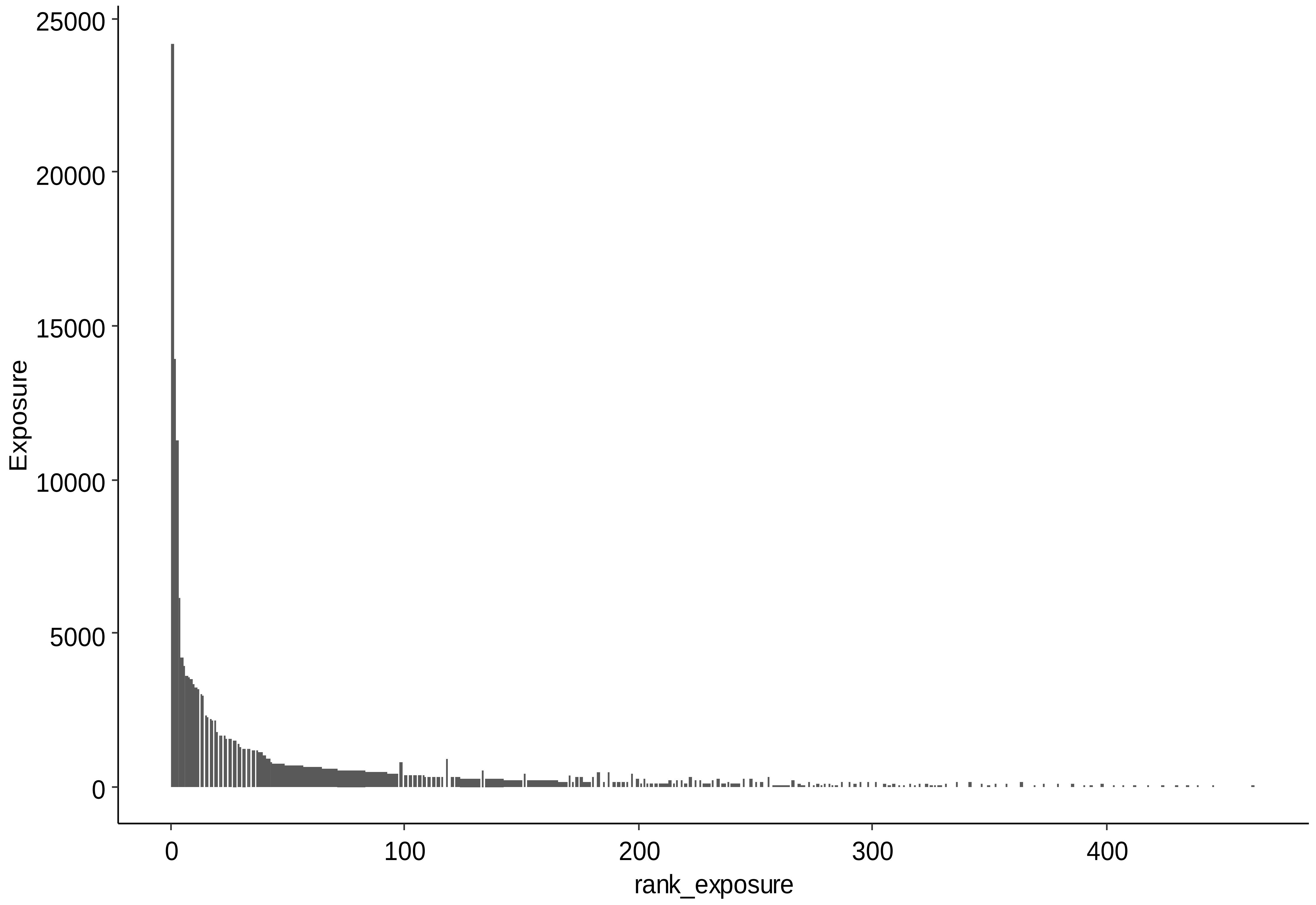
GLMMs offer a solution

- **Generalized Linear Mixed Models (GLMMs) incorporate Bühlmann-Straub credibility to pool individual coefficients towards the mean**

- **A credibility solution looks like**

$$\boldsymbol{\beta_{i,credibility} = z * \beta_i + (1 - z) * \beta}$$





GLMMs offer a solution (2)

- **What is the "mixed" model?**

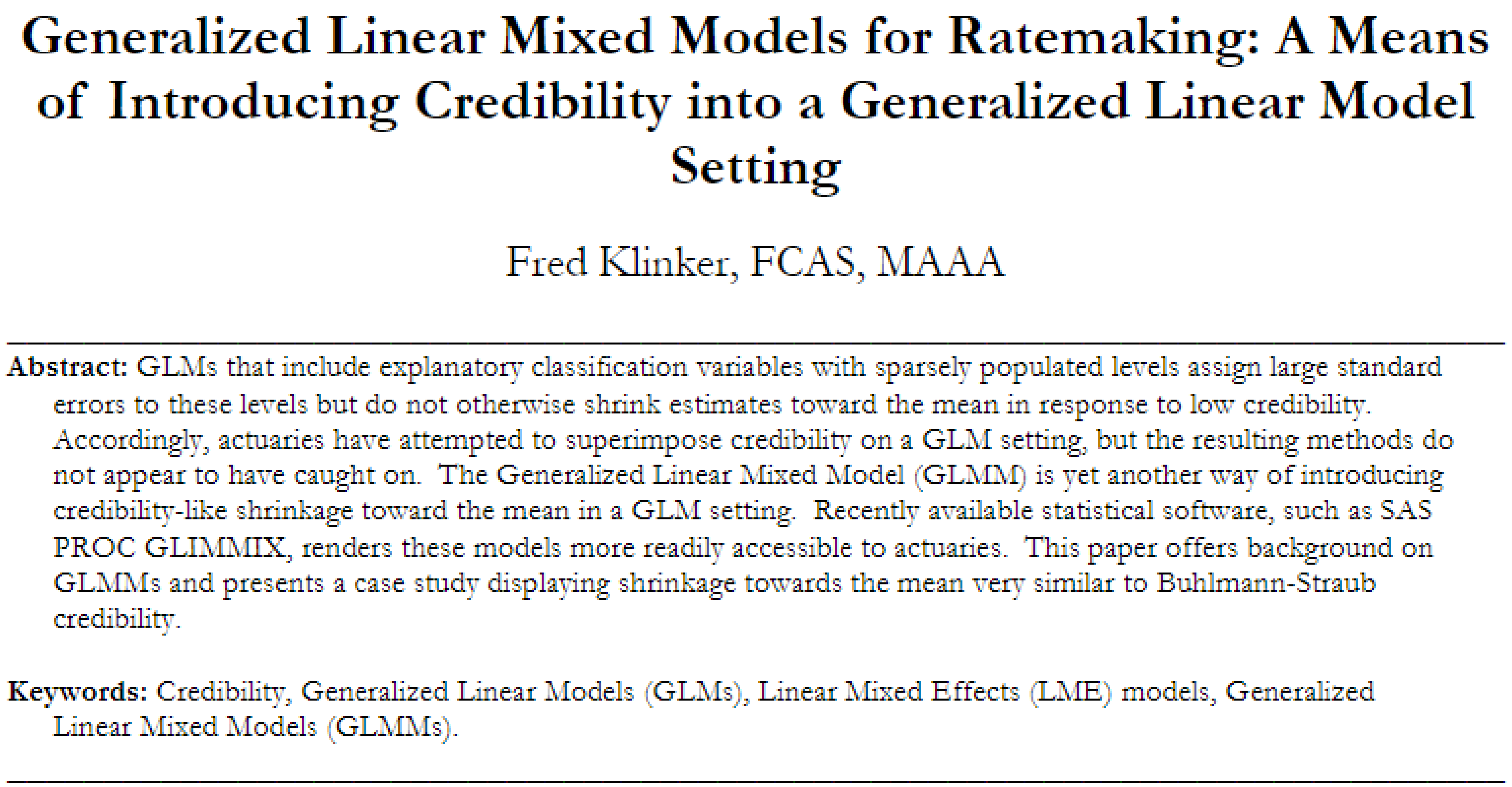
  - GLMMs can include "normal" regression coefficients that do not get a credibility treatment = fixed effects
  - In addition, coefficients with a credibility treatment = random effects
  - The two together = mixed effects model

- **For more, see referenced paper**

- **Disadvantages of GLMM framework:**

  Slow, especially for complex models
  May not always converge

- **How can we incorporate similar ideas into neural network models?**



Generalized Linear Mixed Models for Ratemaking: A Means of Introducing Credibility into a Generalized Linear Model Setting

Fred Klinker, FCAS, MAAA

Abstract: GLMs that include explanatory classification variables with sparsely populated levels assign large standard errors to these levels but do not otherwise shrink estimates toward the mean in response to low credibility. Accordingly, actuaries have attempted to superimpose credibility on a GLM setting, but the resulting methods do not appear to have caught on. The Generalized Linear Mixed Model (GLMM) is yet another way of introducing credibility-like shrinkage toward the mean in a GLM setting. Recently available statistical software, such as SAS PROC GLIMMIX, renders these models more readily accessible to actuaries. This paper offers background on GLMMs and presents a case study displaying shrinkage towards the mean very similar to Buhlmann-Straub credibility.

Keywords: Credibility, Generalized Linear Models (GLMs), Linear Mixed Effects (LME) models, Generalized Linear Mixed Models (GLMMs).

Agenda

- **What are embeddings?**
- **GLMMs**
- **<u>Let’s think Bayesian</u>**
- **Hierarchical models based on embeddings**

GLMMs and Bayesian thinking

- **Bayesian explanation: models generally work by assuming that coefficients follow a normal distribution with mean = 0**

- **Coefficients pulled towards zero unless there is enough credibility for a non-zero value of the MLE derived coefficient**

$$\log f_{\vartheta}(\boldsymbol{Y}, \mathbf{U}) = \ell_{\boldsymbol{Y}}(\vartheta|\mathbf{U}) + \log \pi(\mathbf{U})$$

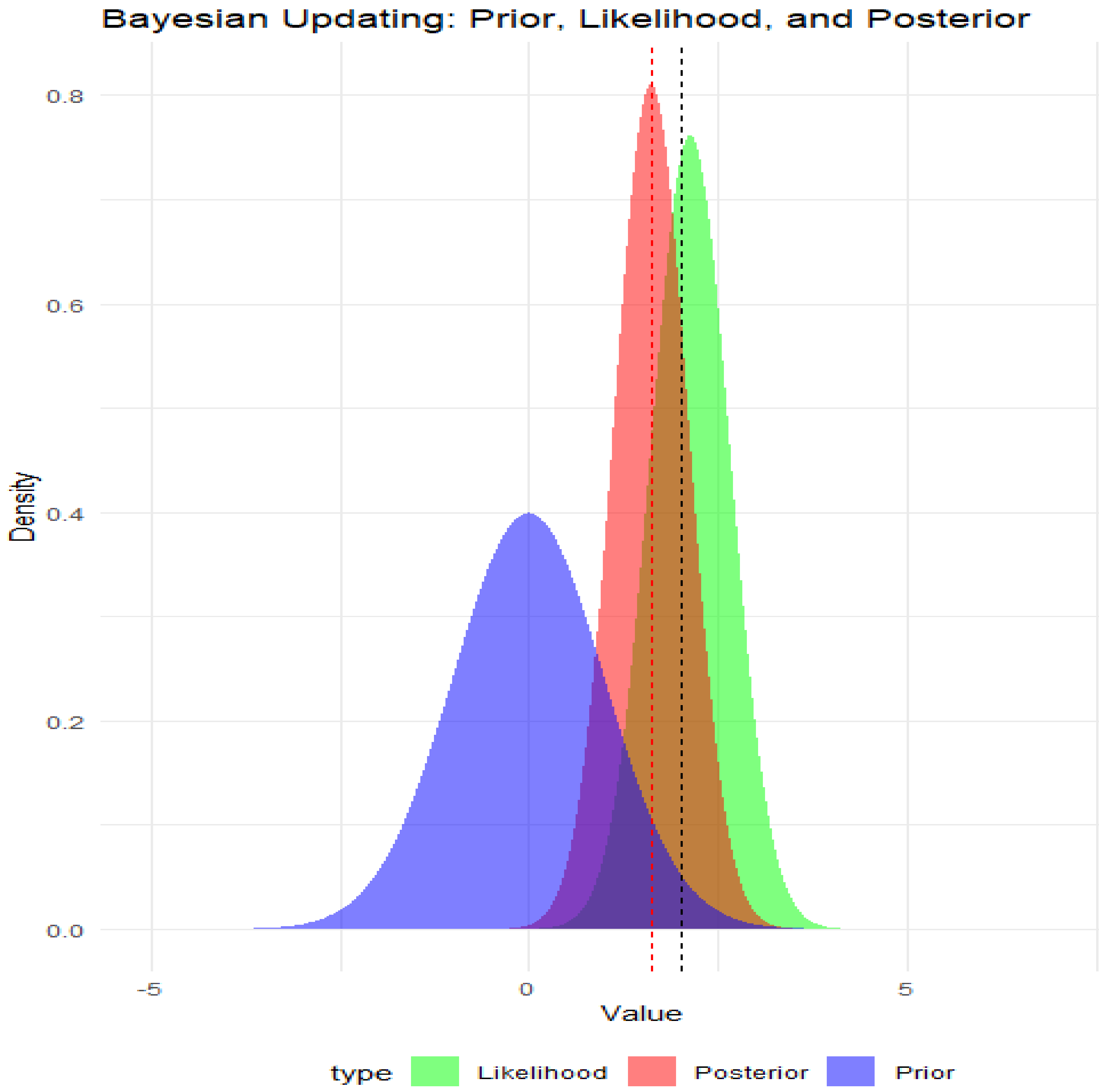
- **Can fit GLMMs using Bayesian methods such as MCMC…**

- **… but these are too slow for complex models such as networks**

- **Closed form solutions (i.e. credibility formula) only available in simple selected cases**

- **Need to rely on approximations to the Bayesian approach**

  - Maximum a posteriori estimation (MAP)
  - Variational inference (VI)



MAP estimation

- **Equivalent to (L2) ridge regularization**

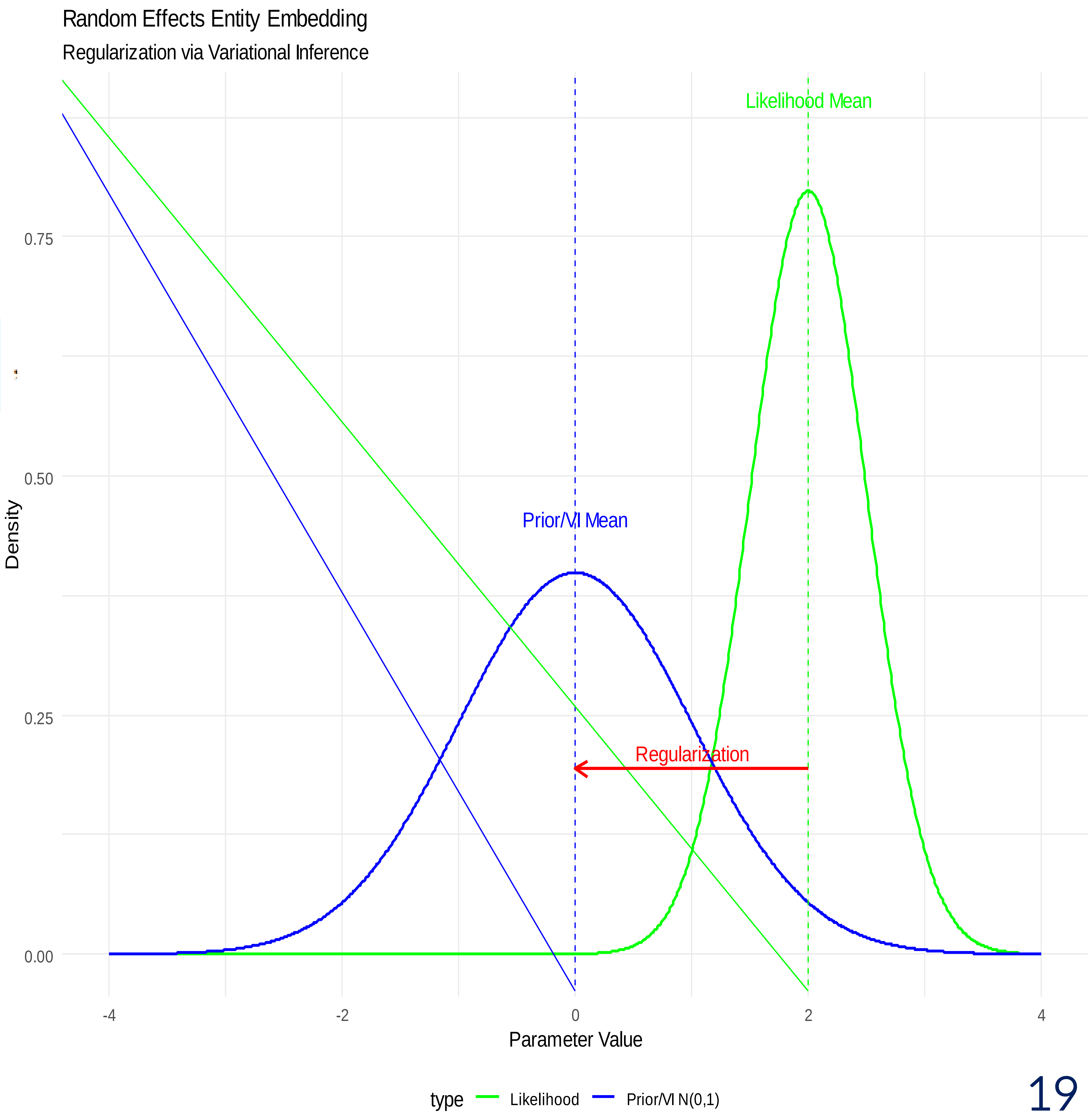
- **Adds a penalty term to the log-likelihood to force coefficients towards to mean of the prior function – which is 0**

$$\frac{1}{\varphi}\sum_{i=1}^{n} v_i \left[Y_i\, h\left(\mathrm{NN}_{\vartheta}(\boldsymbol{x}_i, \mathbf{u}_{j[i]})\right) - \kappa\left(h\left(\mathrm{NN}_{\vartheta}(\boldsymbol{x}_i, \mathbf{u}_{j[i]})\right)\right) - \frac{1}{w_{j[i]}}\,\frac{\varphi}{2\tau^2}\left\|\mathbf{u}_{j[i]}\right\|^2 \right],$$

- <u>**Regularization strength inversely proportional to case weights => low exposure means more regularization**</u>

- **Penalizes large embedding values for rare categories more strongly (towards zero)**

- **Simple to implement and computationally efficient**

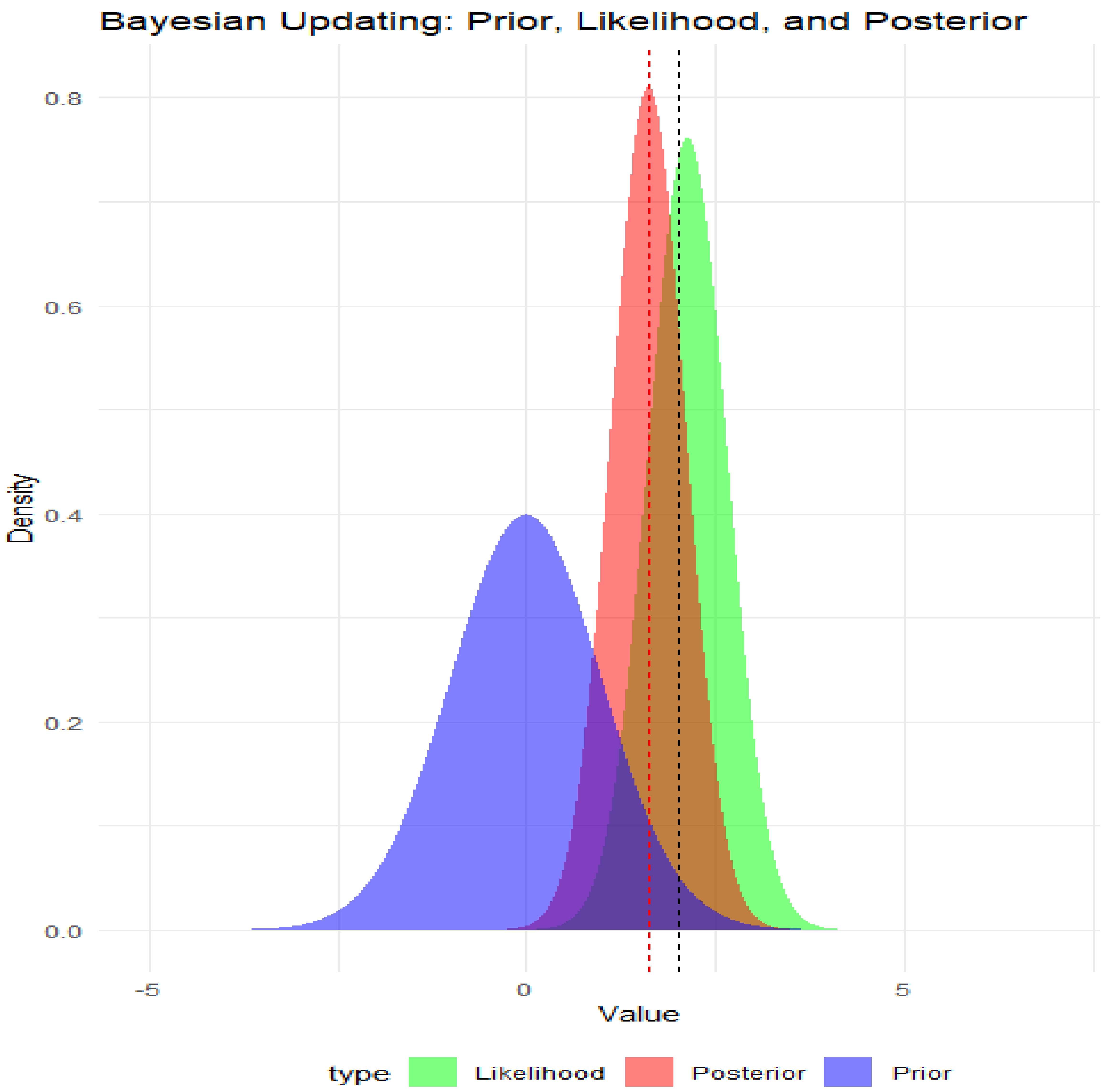


Variational inference

- **Variational inference (VI): Approximate the true intractable posterior distribution by a tractable distribution**
- **This approximation is called variational density**
- **Usually a Gaussian variational density to the posterior is selected (only involves mean and variance parameters)**
- **The quality of the approximation is measured by the Kullback-Leibler (KL) divergence**
- **This leads to a tractable framework which can efficiently be fitted by Monte Carlo sampling and gradient descent**
- **The solution is similar to MAP but one also receive uncertainty estimates**

  MAP is similar to first order Taylor approximation

  VI considers higher order terms



Putting it together

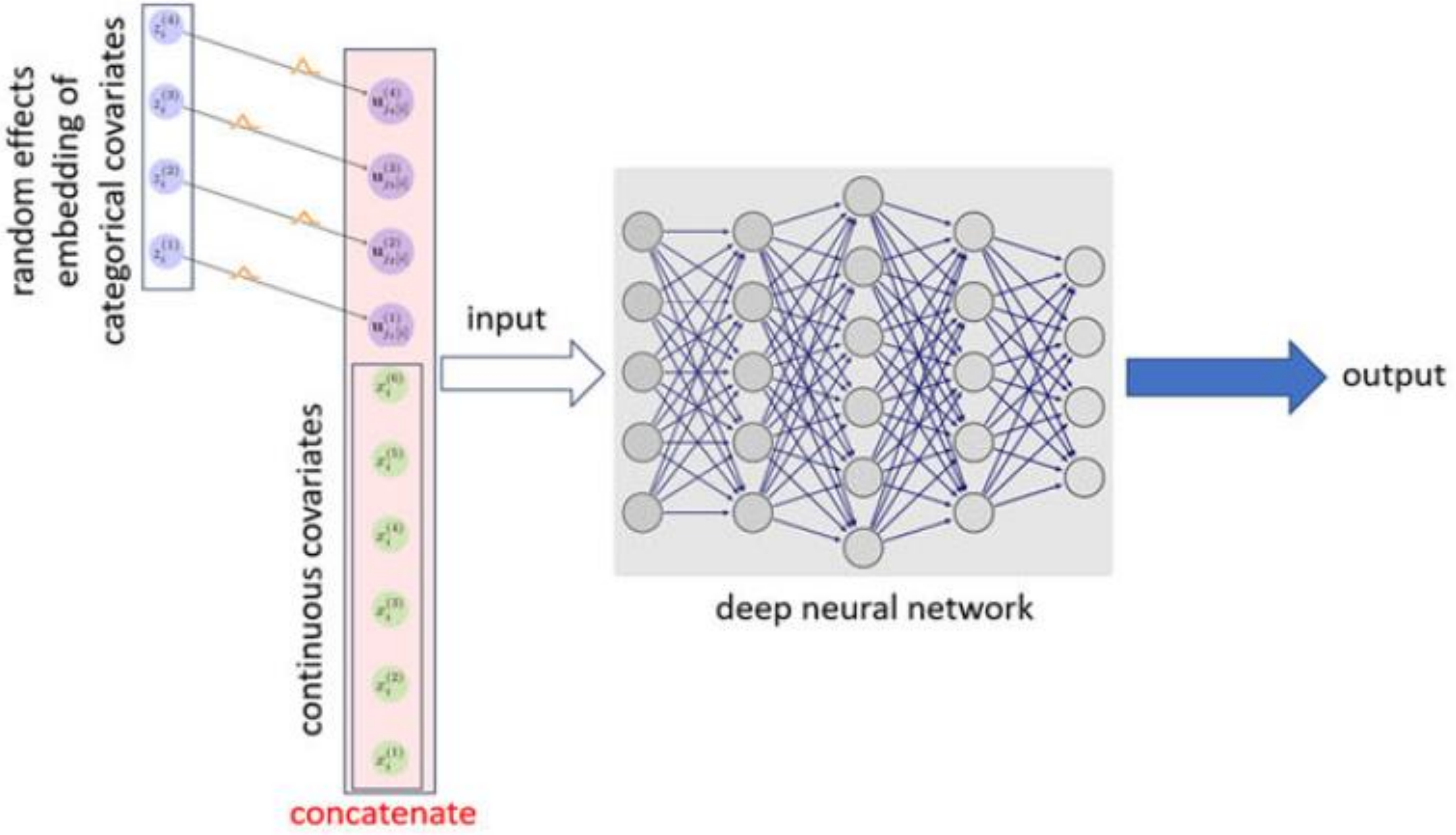
- **Embeddings can be used to capture complex high dimensional information about categorical covariates…**

- **… with the risk of overfitting to the noise, particularly, when there is high cardinality**

- **Developed Bayesian methods to credibility weighted embeddings towards zero, which depend on exposures**

- **Incorporate these into a deep neural network so that both continuous and categorical covariates can contribute to network predictions**

- **Diagram of network used shown on right**

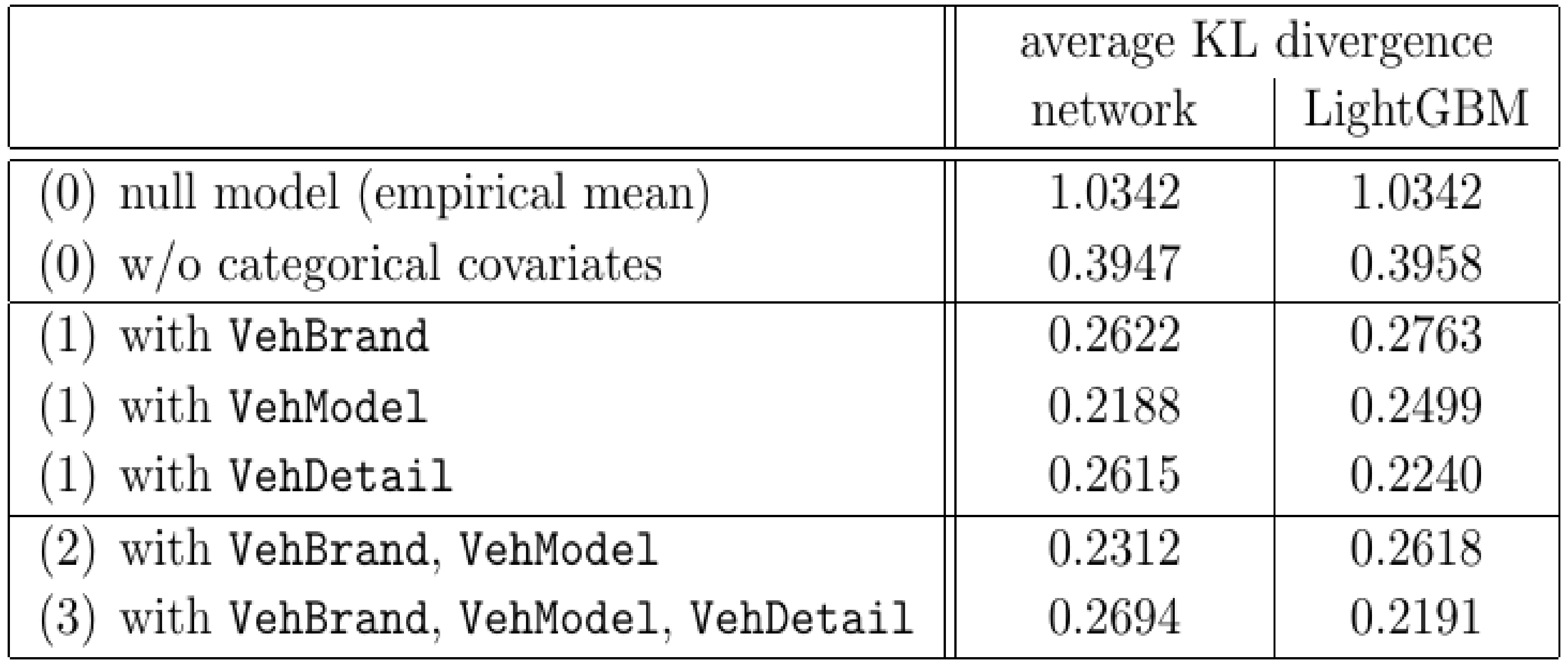


$$\frac{1}{\varphi}\sum_{i=1}^{n} v_i \left[Y_i \, h\left(\mathrm{NN}_{\vartheta}(\boldsymbol{x}_i, \mathbf{u}_{j[i]})\right) - \kappa\left(h\left(\mathrm{NN}_{\vartheta}(\boldsymbol{x}_i, \mathbf{u}_{j[i]})\right)\right) - \frac{1}{w_{j[i]}} \frac{\varphi}{2\tau^2} \left\|\mathbf{u}_{j[i]}\right\|^2 \right],$$

Results

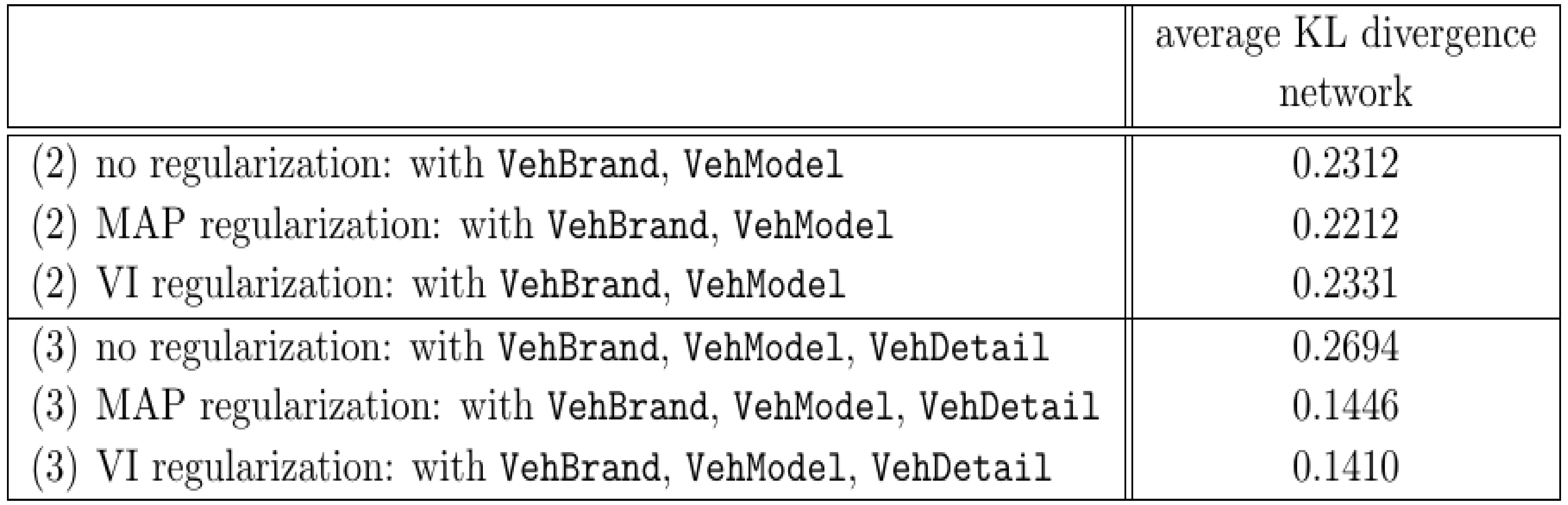
- **Recall that we know the TRUE frequency from the simulated dataset**

- **Fit basic neural network with embeddings (2D) and a LightGBM. Note that network suffers with VehDetail!**



	average KL divergence	
	network	LightGBM
(0) null model (empirical mean)	1.0342	1.0342
(0) w/o categorical covariates	0.3947	0.3958
(1) with `VehBrand`	0.2622	0.2763
(1) with `VehModel`	0.2188	0.2499
(1) with `VehDetail`	0.2615	0.2240
(2) with `VehBrand`, `VehModel`	0.2312	0.2618
(3) with `VehBrand`, `VehModel`, `VehDetail`	0.2694	0.2191

- **Adding regularization to the network produces excellent results; VI slightly better than MAP**



	average KL divergence
	network
(2) no regularization: with `VehBrand`, `VehModel`	0.2312
(2) MAP regularization: with `VehBrand`, `VehModel`	0.2212
(2) VI regularization: with `VehBrand`, `VehModel`	0.2331
(3) no regularization: with `VehBrand`, `VehModel`, `VehDetail`	0.2694
(3) MAP regularization: with `VehBrand`, `VehModel`, `VehDetail`	0.1446
(3) VI regularization: with `VehBrand`, `VehModel`, `VehDetail`	0.1410

Agenda

- **What are embeddings?**
- **GLMMs**
- **Let’s think Bayesian**
- **Hierarchical models based on embeddings**

Hierarchical categorical data

- **Above, we have treated the embeddings representing brand/model/variant of the vehicle independently…**

- **… BUT we know we have a hierarchical structure – cannot have a Toyota model under a Ford make!**

- **How can we exploit this known structure to improve our models?**

- **A good way to think about this is that each level adds a new level of detail to the previous level =>**

Model the incremental information added in each level => cluster embeddings around previous level, e.g., vehicle model clusters around the car brand it belongs to.

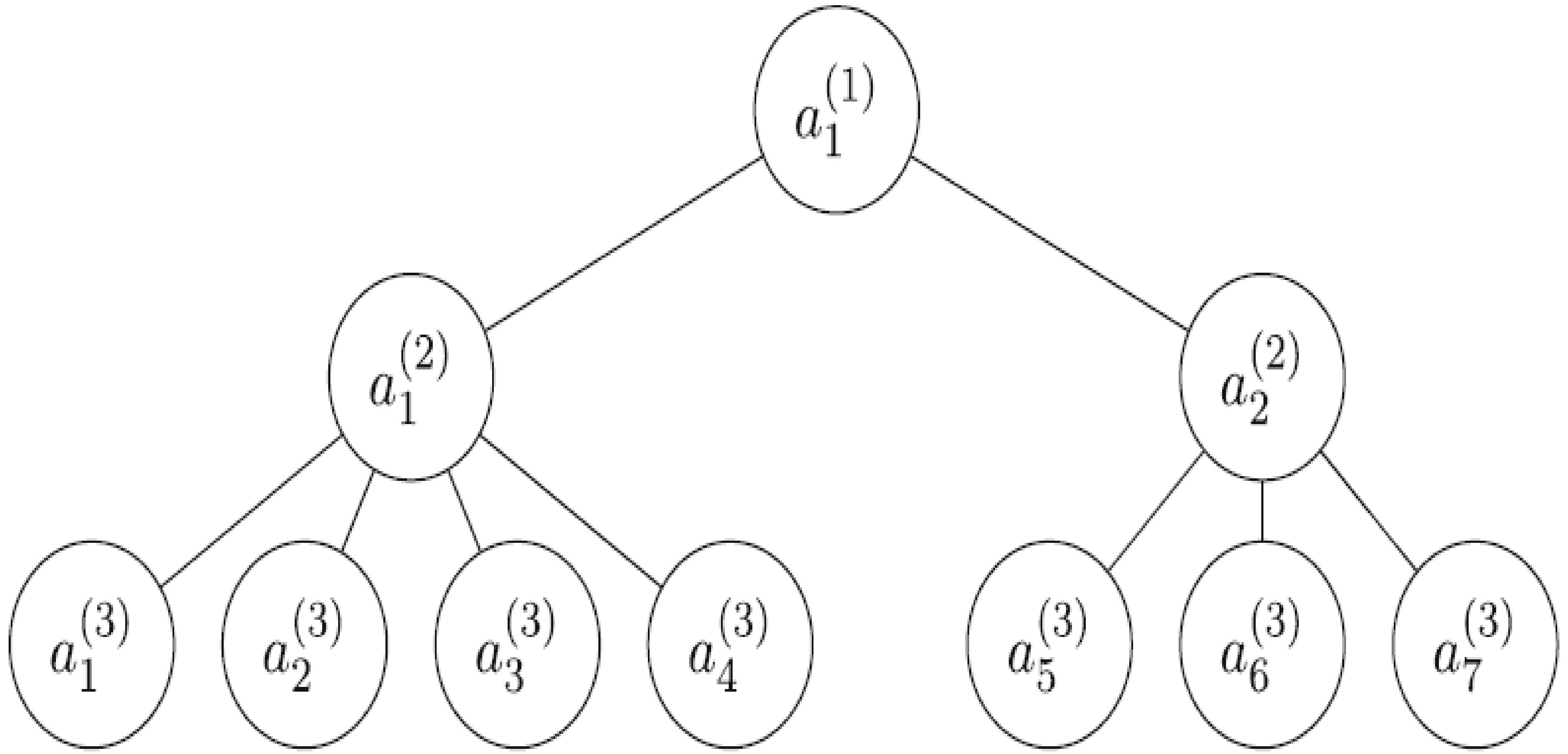


Fig. 3 Descendants of $a_1^{(1)} \in A^{(1)}$ with $T = 3$ generations

$$\Delta^{(t)}_{j'_t} = \mathbf{u}^{(t)}_{j'_t} - \mathbf{u}^{(t-1)}_{j_{t-1}[j'_t]},$$

Models for hierarchical embeddings

- **We have some options for processing hierarchical embeddings:**

- **This has the same structure as a time-series!**

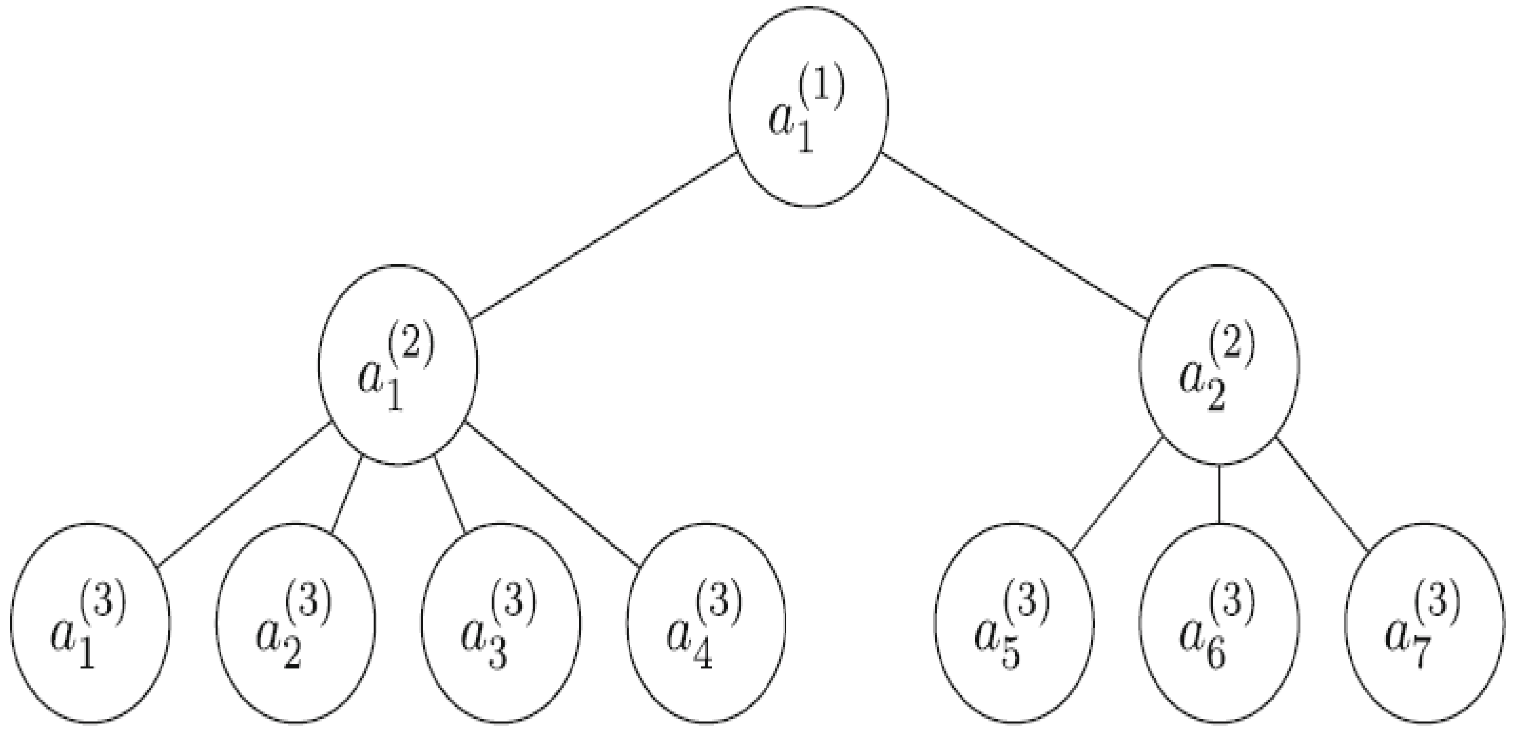
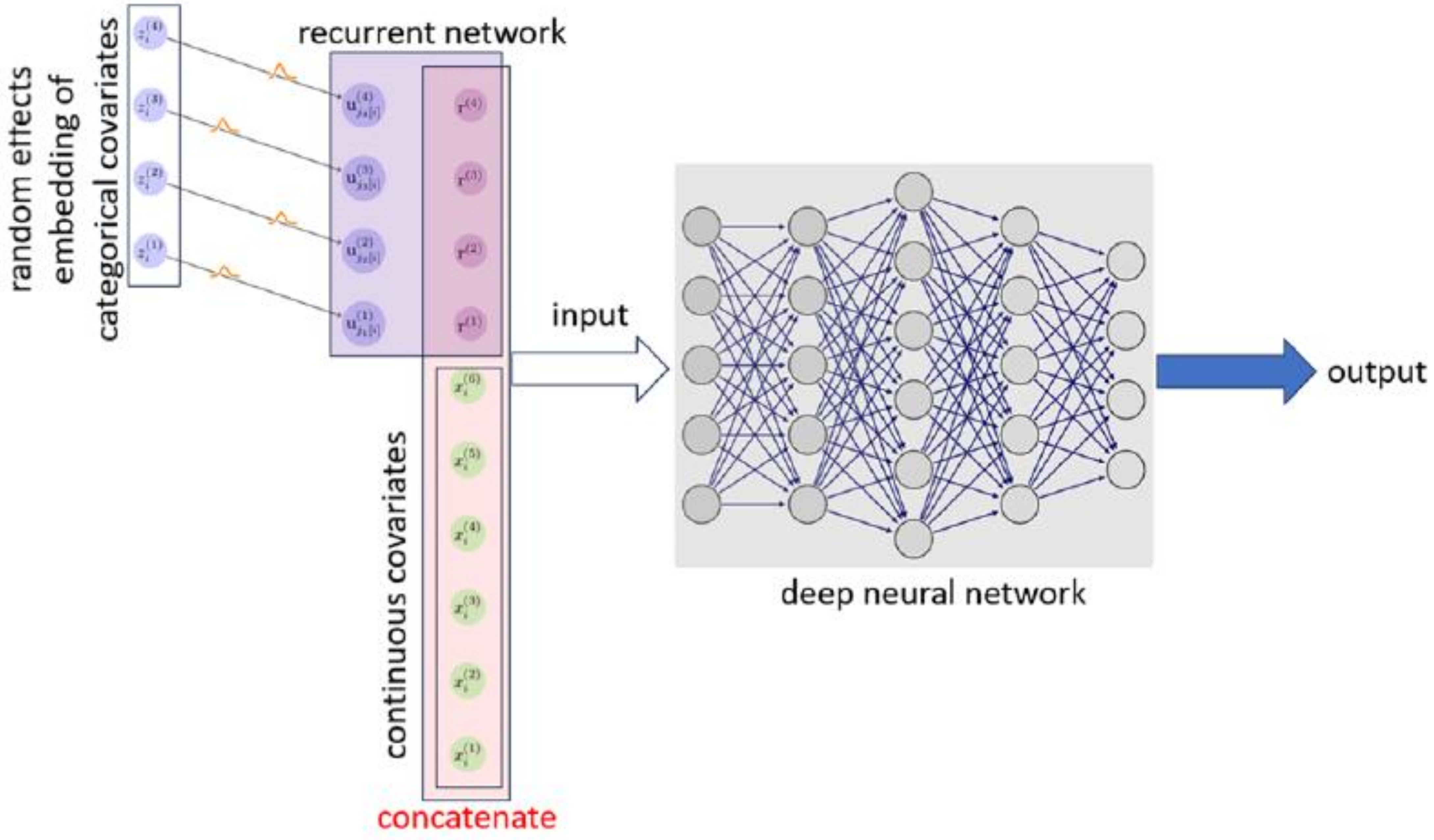


Fig. 3 Descendants of $a_1^{(1)} \in A^{(1)}$ with $T = 3$ generations



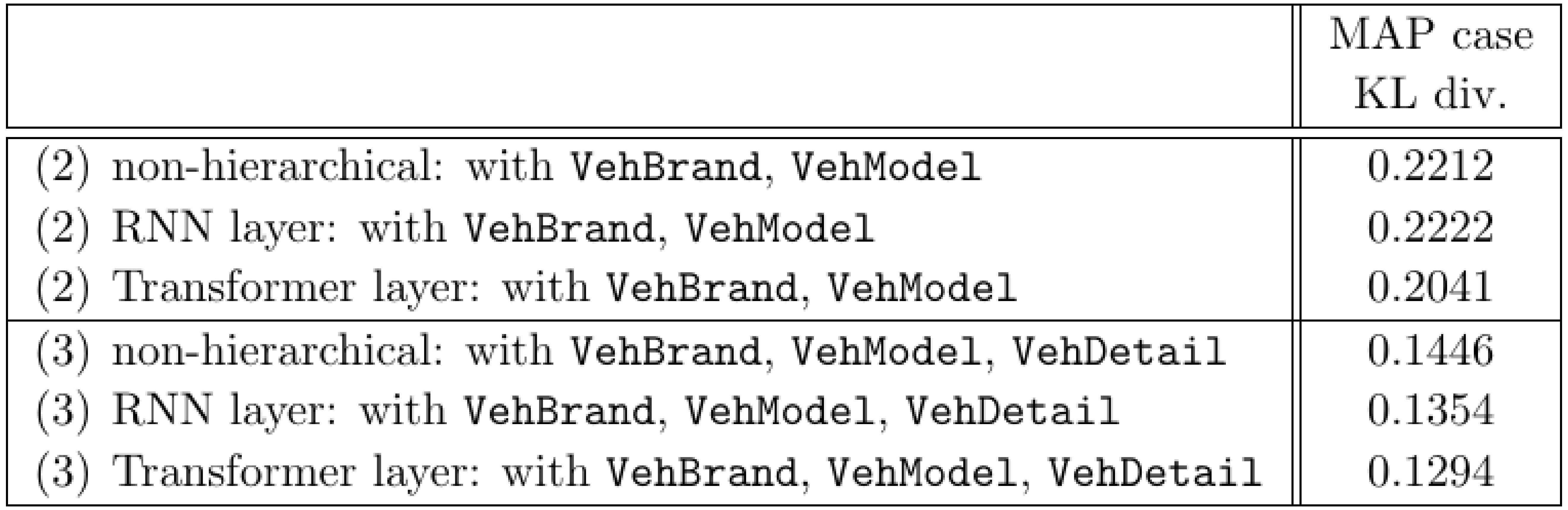
- **(a) Use a recurrent neural network (down the tree)**

- **(b) Use a Transformer (is not causal)**

- **Apply either MAP or VI regularization**

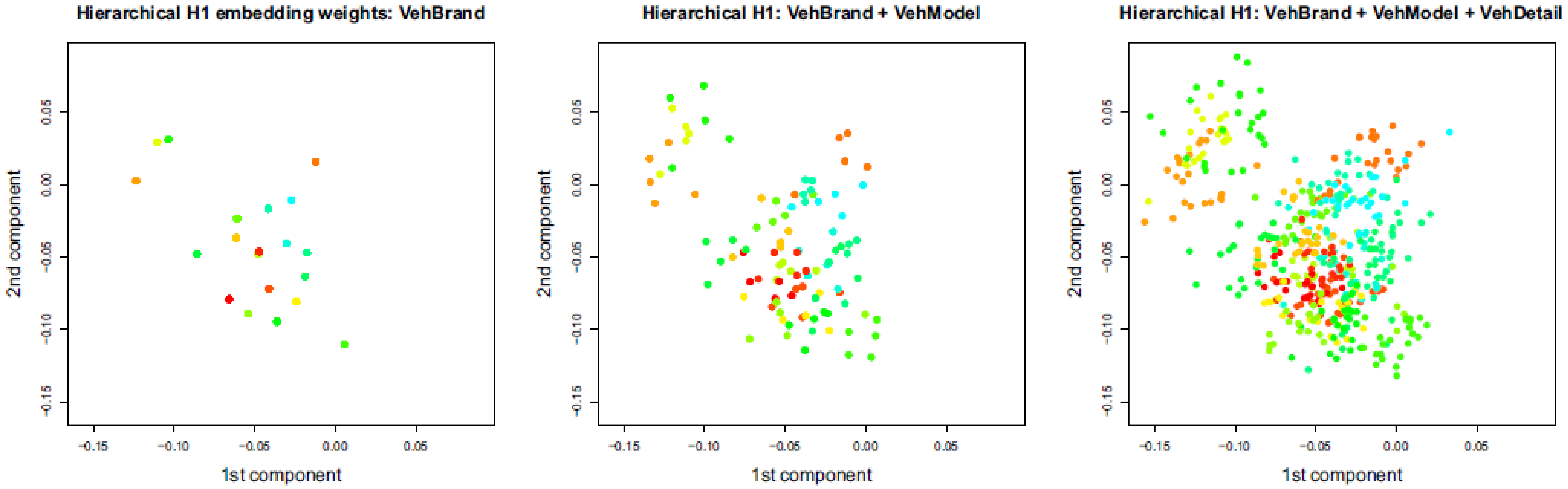
Results

- **Only fit models using MAP as the previous results show VI doesn’t make a huge difference**
- **Results show that incorporating hierarchical approaches improves on previous models!**



	MAP case KL div.
(2) non-hierarchical: with `VehBrand`, `VehModel`	0.2212
(2) RNN layer: with `VehBrand`, `VehModel`	0.2222
(2) Transformer layer: with `VehBrand`, `VehModel`	0.2041
(3) non-hierarchical: with `VehBrand`, `VehModel`, `VehDetail`	0.1446
(3) RNN layer: with `VehBrand`, `VehModel`, `VehDetail`	0.1354
(3) Transformer layer: with `VehBrand`, `VehModel`, `VehDetail`	0.1294

- **Evolution of embeddings**



Summary

- **High-cardinality categorical variables pose significant challenges in regression modeling, particularly in insurance pricing**
- **Entity embedding coupled with regularization techniques offers a powerful solution to handle these variables in neural network models**
- **Two main regularization approaches were explored: MAP estimation and Variational Bayesian inference**
- **Hierarchical structures in categorical variables can be leveraged to further improve model performance**
- **RNN and Transformer architectures show promise in processing hierarchical categorical data**