School of Mathematics and Statistics

The University of Melbourne

# ZAP-GMOE: A $Z$-value covariate-adaptive algorithm with false discovery rate control using Gaussian mixture-of-experts

Callum Holmes

*Supervised by Dr Dennis Leung*

**Abstract.** *Multiple testing entails a challenging trade-off between model flexibility and statistical guarantees for Type I error and power. In settings with few significant observations to discover, recent trends involve incorporating auxiliary information or covariates alongside test statistics to enhance model power. Promising approaches in this field range from lossless data transformations to incorporating partial data masking. However, theoretically optimal procedures still demonstrate better performance on larger datasets with more informative covariates. This paper introduces 'ZAP-GMOE,' an extension of an existing finite-sample Z-value adaptive procedure (ZAP) that applies a regularised Gaussian mixture-of-experts model. We evaluate the effectiveness of this model using both simulated and real datasets. While our results do not demonstrate a competitive improvement in performance over related methods, this investigation offers insights into the limitations and future areas of research for extending the ZAP procedure.*

*Thesis submitted as part of the Master of Science (Mathematics & Statistics)*

May 2023

*Acknowledgement*

I would like to express my gratitude to my supervisor Dr Dennis Leung for their support and guidance throughout my project and in writing this paper. I also wish to thank my family for their support of my studies throughout my life, without which writing this paper would not have been possible, and my friends who have encouraged my endeavours and provided a space of mental sanctuary from study.

This research was supported by The University of Melbourne's Research Computing Services and the Petascale Campus Initiative.

# Notation

| | |
|---|---|
| $:=$ | Variable definitions |
| $\equiv$ | Equivalent definitions |
| $[n]$ | Given $n \in \mathbb{N}$, the integer set $\{1, \dots, n\}$ |
| $\in_m$ | Measurability, e.g. $X \in_m \mathcal{F}$ |
| $\phi(x \,\vert\, \mu, \sigma^2)$ | The probability density of a normal distribution with mean $\mu$ and variance $\sigma^2$, i.e. $\phi(x \,\vert\, \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\big(-\frac{1}{2\sigma^2}(x-\mu)^2\big)$ |
| $\Phi(x \,\vert\, \mu, \sigma^2)$ | The cumulative density function for a normal distribution with mean $\mu$ and variance $\sigma^2$, i.e. $\Phi(x \,\vert\, \mu, \sigma^2) = \int_{-\infty}^{x} \phi(t \,\vert\, \mu, \sigma^2) dt$ |
| $\phi(x), \Phi(x)$ | Shorthand for $\phi(x \,\vert\, \mu, \sigma^2)$, $\Phi(x \,\vert\, \mu, \sigma^2)$ |
| $\Vert\boldsymbol{x}\Vert$ | $\sqrt{\boldsymbol{x}^T\boldsymbol{x}}$, the euclidean norm |

# Contents

# Chapter 1

# Preliminaries

## 1.1. Introduction

In multiple testing, the more tests that are performed the more likely it is that some tests will be significant purely by chance, even if there is no real effect or association. Consequently, experimenters are tasked with modifying their testing procedure to control the frequency of false discoveries across the entire testing suite.

One recognised strategy for controlling the occurrence of false discoveries is to limit the *False Discovery Rate* (FDR). First introduced in the influential [BH95], the definition is as follows: if $\mathbb{V}$-many true null hypotheses are rejected (*False Discoveries/Positives*), and $\mathbb{S}$-many true null hypotheses are not rejected (*True Discoveries/Positives*), the FDR is the expected value of the *False Discovery Proportion* (FDP):

$$\mathrm{FDR} = \mathbb{E}\left[\mathrm{FDP}\right] = \begin{cases} \mathbb{E}\left[\frac{\mathbb{V}}{\mathbb{V}+\mathbb{S}}\right] & \mathbb{V}+\mathbb{S} > 0 \\ 0 & \text{otherwise} \end{cases}$$

Relative to other Bonferroni-type multiple comparisons procedures at the time, this novel rank-and-screening procedure was more powerful in settings where the number of hypotheses $m$ was large, and where there were few non-null hypotheses to discover.

Recent works in FDR inference develop covariate adaptive procedures, which leverage side information or covariates to boost statistical power while allowing analysts to

interact with the data. This is especially useful in settings with few non-null signals to discover, where additional information can support otherwise insignificant statistics. The power of an adaptive approach lies in its ability to leverage highly contextual information - for example, [CF21] permits an analyst to freely interrogate the censored dataset during execution and change the direction of the algorithm, while guaranteeing FDR control. Many interactive approaches exist, reflecting the various data relationships and contexts that analysts can encounter - these approaches can involve data masking as in [CF21] but can also include grouping $p$-values and applying a weighted BH procedure ([Ign+16]), or using augmented data generated through "knockoff filters" ([BC16]).

Covariate-based multiple testing is highly applicable in the biomedical fields. In genome-wide association studies (GWAS) for example, FDR-controlling procedures are powerful in identifying sections of the genome (single nucleotide polymorphisms (SNPs)) associated with traits of interest - examples of this application are included in [Yur+20]) (which applies the AdaPT model from [LF16]) and the comparative analysis in Section 4.3 of [CF21]. Another application is in RNA sequencing, where we can test for gene expressions by comparing read counts across two experimental environments, which has been explored in the context of hippocampal cellular dissociation in [Har+19] and analysed in [LS21].

Our interest lies in making inference on the $Z$-values, rather than with $p$-values that can be computed with lossy transformations in two-sided or interval testing (e.g. [CF21], [CSW19]). The authors of [LS21] explicitly cite these concerns[1] as a motivation for their $Z$-value covariate-adaptive procedures (ZAP), arguing that applying a non-bijective transformation not only discards information in the $Z$-values, but corrupts the relationship between these test statistics and the available side information. Their paper provides a finite-sample ZAP procedure that uses an underlying beta-mixture as a data model, as well as an asymptotic procedure with asymptotic FDR control, which both illustrate strong performance compared to other covariate-adaptive methods including

[1] See Section 2.2 in [LS21]

the more recent [CF21].

However, there are performance gaps highlighted in [LS21] worth exploring. Comparing the performance of ZAP, AdaPT, AdaPT-GMM$_g$ and other competitive approaches to provably bayes-optimal FDR-control procedures, the authors identified an approximately 6% power gap in the True Positive Rate (TPR).[2] This gap was strongest on data distributions where the distribution of the $Z$-values tended to diverge significantly in proportion to the covariates.

This paper attempts to close this performance gap on datasets with highly informative covariates by extending the finite-sample ZAP procedure. This extension incorporates a more flexible Gaussian Mixture-of-Experts (GMoE) working model for the $Z$-values with theoretical guarantees on FDR control, which we call "ZAP-GMOE".

The rest of this Chapter presents a formal description of the target problem, and relevant background on the GMoE model family. Chapter 2 offers a formal outline of the ZAP-GMOE procedure, and the efficacy of this model is explored through experiments in Chapter 3 in comparison to ZAP, AdaPT-GMM$_g$ and other existing methods.

## 1.2. Formal Problem Definition

Consider a suite of $n$ statistical tests that are conducted simultaneously, where the $i$th test instance produces a test statistic $Z_i$, and each are in relation to some effect size $\mu_i \in \mathbb{R}$. Also available is a set of $p$-dimensional covariate feature vectors corresponding to each instance $\{\boldsymbol{x}_i\}_{i=1}^n$, which for notational simplicity can be summarised in a $n \times p$ matrix $X := \begin{bmatrix} \boldsymbol{x}_1 & \boldsymbol{x}_2 & \dots & \boldsymbol{x}_n \end{bmatrix}^T$. We are interested in discerning about the value of each $\mu_i$, and in particular want to test the veracity of the $n$ null and alternative hypotheses

$$H_{i,0} : \mu_i = 0 \quad \text{vs} \quad H_{i,1} : \mu_i \neq 0 \quad i \in [n] \tag{1.1}$$

[2] The authors have recently updated their paper and package, resulting in a decrease of this gap to less than 2%. However, the gap is still persistent across the same family of datasets.

Denoting the set of true null hypotheses as $\mathcal{H}_0 := \{i \in [n] : \mu_i = 0\}$, and true alternatives or "discoveries" as $\mathcal{H}_1 = [n] \setminus \mathcal{H}_0$, informally our objective is to choose a set of rejections $\mathcal{R} \subset \{H_{i,0} : i \in [n]\}$ with as many hypotheses in $\mathcal{H}_1$ as possible, while controlling the proportion of rejected hypotheses from $\mathcal{H}_0$.

A reasonable proxy for model power is to look at the expected recall of the model, also known as the True Positive Rate (TPR). Using the indicators $\mathbb{H}_i := \mathbb{I}(\mu_i \neq 0)$, the TPR is defined as

$$\text{TPR} = \mathbb{E}\left[\frac{\sum_{i=1}^{n} (\mathbb{H}_i \times \mathbb{I}(H_{i,0} \in \mathcal{R}))}{\left(\sum_{i=1}^{n} \mathbb{H}_i\right) \vee 1}\right]$$

This paper seeks a model that selects rejections using the dataset $\mathcal{D} = \{(Z_i, \boldsymbol{x}_i)\}_{i=1}^n$ that maximises the TPR with FDR control at some fixed level $\alpha \in (0, 1)$. As in [LS21], we assume that the tuples $(\mathbb{H}_i, Z_i, \boldsymbol{x}_i)_{i\in[n]}$ are i.i.d. and that the data follow the two-group mixture model

$$(Z_i \,|\, \boldsymbol{x}_i) \sim f(Z \,|\, \boldsymbol{x}) = (1 - \pi(\boldsymbol{x}))\phi(Z) + \pi(\boldsymbol{x}) f_{1,\boldsymbol{x}}(Z)$$

This model assumes that each data has a conditional probability $\pi(\boldsymbol{x}_i) := \mathbb{P}(\mathbb{H}_i = 1 \,|\, \boldsymbol{x}_i)$ of being generated from the alternative conditional density $f_{1,\boldsymbol{x}}(Z)$, and a conditional probability $1 - \pi(\boldsymbol{x}_i)$ of being standard normal, the null distribution. We also assume that the $\pi(\boldsymbol{x}_i)$ are generally small, and the dataset has few non-null signals to discover.

## 1.3. Flexibility in Mixture-of-Experts Models

Finite mixture models as defined in [MP04] are a general family of models that, in their basic form, model a continuous density $f(z_i)$ for each instance as a linear combination of component densities, i.e.

$$f_Z(z) = \sum_{k=1}^{K} \pi_k f_k(z)$$

where the $\pi_k$ are mixing proportions such that $\sum_{k=1}^{K} \pi_k = 1$ and $0 \leq \pi_k \leq 1$. Specifically for observations $\{Z_i\}_{i=1}^n$ paired with features $\{\boldsymbol{x}_i\}_{i=1}^n$, mixture-of-experts (MoE) models

are an extension of this model introduced in [Jac+91]. In a general MoE model, a dataset $\mathcal{D} = \{(\boldsymbol{x}_i, Z_i)\}_{i=1}^n$ is generated by one of $K$ expert densities with probability $\pi_k$ (also called a mixing proportion) - these are assumed distinct a priori.

The key difference of the MoE model is that both the mixing proportions and expert densities are functions of $\boldsymbol{x}_i$ - as a result, an MoE can offer a better approximation for data with distinct clusters of behaviour, as suggested in [YWG12]. Moreover, the authors in [NLM16] formally establish this flexibility, by demonstrating that the MoE family of functions taking feature vectors $\boldsymbol{x}$ from a compact set $\mathbb{X}$ can uniformly approximate any continuous function on the domain $\mathbb{X}$ up to an arbitrary precision.

To express this model, one can define a latent variable annotating which expert distribution is used for each $Z_i$, i.e.

$$(\Gamma_i \,|\, \boldsymbol{x}_i, Z_i) \sim \text{Multinom}\,(1, \{\pi_{i,1}, \ldots, \pi_{1,K}\}) \tag{1.2}$$

where

$$\sum_{k=1}^{K} \pi_{i,k} = 1 \quad \forall\, i \in [n] \tag{1.3}$$

The mixing probabilities $\pi_{i,k}$ for each expert are determined by the softmax functions

$$\pi_{i,k} \equiv \pi_k(\boldsymbol{x}_i \,|\, w_{k,0}, \boldsymbol{w}_k) = \frac{\exp\big(w_{k,0} + \boldsymbol{w}_k^T \boldsymbol{x}_i\big)}{1 + \sum_{l=1}^{K-1} \exp(w_{l,0} + \boldsymbol{w}_l^T \boldsymbol{x}_i)}, \quad k \in [K-1] \tag{1.4}$$

and (for $k = K$)

$$\pi_{i,K} = \frac{1}{1 + \sum_{l=1}^{K-1} \exp(w_{l,0} + \boldsymbol{w}_l^T \boldsymbol{x}_i)} \tag{1.5}$$

where we have imposed that $w_{K,0} = 0,\ w_K = \mathbf{0}$ to respect Eq (1.3).

For notational convenience in this paper, we summarise the set of $K-1$ free intercepts and coefficients with the vector $\boldsymbol{w} = (w_{1,0}, \boldsymbol{w}_1^T, \ldots, w_{K-1,0}, \boldsymbol{w}_{K-1}^T)^T$. This element of the model is referred to as the *gating network* in machine learning literature.

Given a choice of expert $\Gamma_i$, each predictor is then assumed to be generated from a corresponding conditional expert distribution:

$$(Z_i \,|\, \Gamma_i, \boldsymbol{x}_i) \sim f_{Z\,|\,\Gamma}(Z_i \,|\, \Gamma_i, \boldsymbol{x}_i) =: f_k(Z_i \,|\, \boldsymbol{x}_i),$$

This set of component distributions $\{f_k(Z_i \,|\, \boldsymbol{x}_i)\}_{k=1}^K$ is also known in literature as the *expert network*, and their distribution can vary across applications, though are generally linear in each $\boldsymbol{x}_i$. A key example is the Gaussian MoE (GMoE), which assumes Gaussian conditionals with data-dependent means:

$$(Z_i \,|\, \Gamma_i = k, \boldsymbol{x}_i) \sim \mathrm{N}\big(\beta_{k,0} + \boldsymbol{\beta}_k^T \boldsymbol{x}_i, \sigma_k^2\big) \tag{1.6}$$

Again for convenience, we abbreviate the mean intercepts and coefficients with $\boldsymbol{\beta} := \big(\beta_{1,0}, \boldsymbol{\beta}_1^T, \ldots, \beta_{K,0}, \boldsymbol{\beta}_K^T\big)^T$, and encapsulate all model parameters with the vector $\boldsymbol{\theta} := \begin{pmatrix} \boldsymbol{w}^T & \boldsymbol{\beta}^T & \sigma_1^2 & \ldots & \sigma_K^2 \end{pmatrix}^T$.

Given the gating and expert networks, the marginal density for each data can be expressed as

$$f(Z_i \,|\, \boldsymbol{x}_i) = \sum_{k=1}^K f_{Z \,|\, \Gamma}(Z_i \,|\, \boldsymbol{x}_i, \Gamma_i = k) \mathbb{P}(\Gamma_i = k \,|\, \boldsymbol{x}_i) = \sum_{k=1}^K \pi_{i,k} f_k(Z_i \,|\, \boldsymbol{x}_i) \tag{1.7}$$

MoE models have been extended to effectively model larger datasets with higher-dimensional feature vectors. The authors of [HC19] achieve this in an MLE setting by encouraging sparsity, imposing Lasso penalties on the expert and gating parameters $\boldsymbol{w}, \boldsymbol{\beta}$. Explicitly, the authors opt to maximise the penalised log-likelihood

$$\log PL(\boldsymbol{\theta}) = \log L(\boldsymbol{\theta}) - \sum_{k=1}^K \lambda_k \|\beta_k\|_1 - \sum_{k=1}^{K-1} \gamma_k \|\boldsymbol{w}_k\|_1 \tag{1.8}$$

where $\log L(\boldsymbol{\theta}) = \log \prod_{i=1}^n f(Z_i \,|\, \boldsymbol{x}_i)$ is the observed-data log-likelihood, $\gamma_k \geq 0,\ k \in [K-1]$ are the gating network penalties, and $\lambda_k \geq 0,\ k \in [K]$ are the expert network penalties. It is the flexibility of this MoE model and its robustness in sparser covariate spaces that has motivated the inception of the ZAP-GMOE model. In this paper we take the core machinery of the finite-sample ZAP procedure in [LS21], but use a regularised GMoE data-driven probability model to replace its beta-mixture model, in an attempt to improve performance over the original ZAP procedure in sparser, high-dimensional data settings.

# Chapter 2

# ZAP Gaussian Mixture-of-Expert Model

## 2.1. Model outline

The proposed ZAP-GMOE algorithm is an EM-based algorithm with the rank-and-reveal structure of the finite-sample ZAP algorithm introduced in [LS21]. Replacing their beta-mixture model, we utilise a $Z$-value GMoE working model based on Eq (1.6) and Eq (1.7) to estimate the marginal densities:

$$\begin{aligned}(Z_i \,|\, \boldsymbol{x}_i) \sim f(Z_i \,|\, \boldsymbol{x}_i, \boldsymbol{\theta}) &= \sum_{k=1}^{K} f(Z_i \,|\, \boldsymbol{x}_i, \Gamma_i = k)\mathbb{P}(\Gamma_i = k \,|\, \boldsymbol{x}_i) \\ &= \sum_{k=1}^{K} \pi_{i,k}\phi\left(Z_i \,|\, \beta_{k,0} + \beta_k^T \boldsymbol{x}_i, \sigma_k^2\right) \end{aligned} \tag{2.1}$$

Here, the $\pi_{i,k}$ are determined by the softmax functions in Eq (1.4) and Eq (1.5).

At the $t$th iteration of ZAP-GMOE, a given subset $\mathcal{M}_t$ of the data is masked in accordance with the procedure in Section 2.2, yielding the masked dataset $\mathcal{D}_t = \left\{(\boldsymbol{X}_i, \widetilde{Z}_{t,i})\right\}_{i=1}^{n}$. Conditioning on this, we update the working model parameters ($\boldsymbol{\beta}$) with an EM algorithm; this step is summarised in Algorithm 2 and its implementation details are provided in Appendix A.3.

**Algorithm 1:** ZAP-GMOE

**Data:** $\mathcal{D} = (Z_i, \boldsymbol{x}_i)_{i=1}^n$

**Model Hyper-parameters:** $\alpha > 0, K \in \{2, \dots\}, \{\lambda_k\}_{k\in[K]}, \{\gamma_k\}_{k\in[K-1]}$

**Masking Hyper-parameters:** $0 < \alpha_m \leq \lambda_m < \nu \leq 1$

**Model Initialisation:** $\hat{\boldsymbol{\theta}} \leftarrow \hat{\boldsymbol{\theta}}_0,\ \mathcal{M}_0 \leftarrow \{i \in [n] : m_i < \alpha_m\}$

**for** $t = 0, 1, \dots$ **do**

- $\widehat{\mathrm{FDP}}_t \leftarrow \frac{1+|\mathcal{A}_t|}{\zeta|\mathcal{R}_t|}$
- **if** $\widehat{FDP}_t \leq \alpha$ **then**
  - **return** *Rejection set* $\{H_{i,0} : i \in \mathcal{R}_t\}$
- **if** $t \mod \lfloor n/nfits \rfloor = 0$ **then**
  - `// (Fit model` $'nfits'$ `times)`
  - $\hat{\boldsymbol{\theta}} \leftarrow$ `EM-update` $\left((\boldsymbol{x}_i, m_i, s_i)_{i\in\mathcal{M}_t}, (\boldsymbol{x}_i, Z_i)_{i\notin\mathcal{M}_t}, \hat{\boldsymbol{\theta}}\right)$
- $\mathcal{M}_{t+1} \leftarrow \mathcal{M}_t \setminus \left\{\arg\max_{i\notin\mathcal{M}_t}$ `Q-estimate`$((\boldsymbol{x}_i, Z_i), \hat{\boldsymbol{\theta}})\right\}$
- $\mathcal{A}_{t+1} \leftarrow \{i \in \mathcal{M}_{t+1} : p_i \in [\lambda_m, \nu]\}$
- $\mathcal{R}_{t+1} \leftarrow \{i \in \mathcal{M}_{t+1} : p_i \in [0, \alpha_m]\}$

**end for**

With the fitted model, we then rank the masked instances $\mathcal{M}_t$ using the *assessor function* (Eq (2.6), Section 2.3) and the highest-ranking instance is unmasked.

The overall logic of the ZAP-GMOE procedure is summarised in Algorithm 1, and we formalise its FDR control in Theorem 2.1 below.

**Theorem 2.1 (ZAP-GMOE FDR Control).** *Assume that the null p-values are mutually independent, and independent of the non-null p-values, and further that the null p-values have non-decreasing density. Then the ZAP-GMOE procedure in Algorithm 1 controls the FDR at the target level* $\alpha$.

A proof of this theorem is available in Appendix A.1.

## 2.2. Masking

The objective of the masking procedure is to reveal partial information about the true $Z_i$ to the algorithm, so that the algorithm can leverage some information to make rejections, but without compromising the FDR control. The motive for this procedure is also discussed in Section 2.2.1. To mask the subset $\mathcal{M}_t \subset [n]$ of instances, we follow the procedure outlined in Appendix C of the AdaPT-GMM$_g$ paper [CF21]. In the $t$th iteration of the algorithm, to mask an instance $(Z_i, \boldsymbol{x}_i),\ i \in \mathcal{M}_t$, we first compute an associated two-tailed $p$-value

$$p_i = q(Z_i) := 2(1 - \Phi(|Z|)) \tag{2.2}$$

Secondly, we compute the masked value $m_i = g(p_i)$, using the asymmetric "tent" $p$-value masking function introduced in [CF21]

$$g : [0,1] \to [0,1],\ g(p) := \begin{cases} \frac{\nu - p}{\zeta} & p \in [\lambda_m, \nu] \\ p & \text{otherwise} \end{cases} \tag{2.3}$$

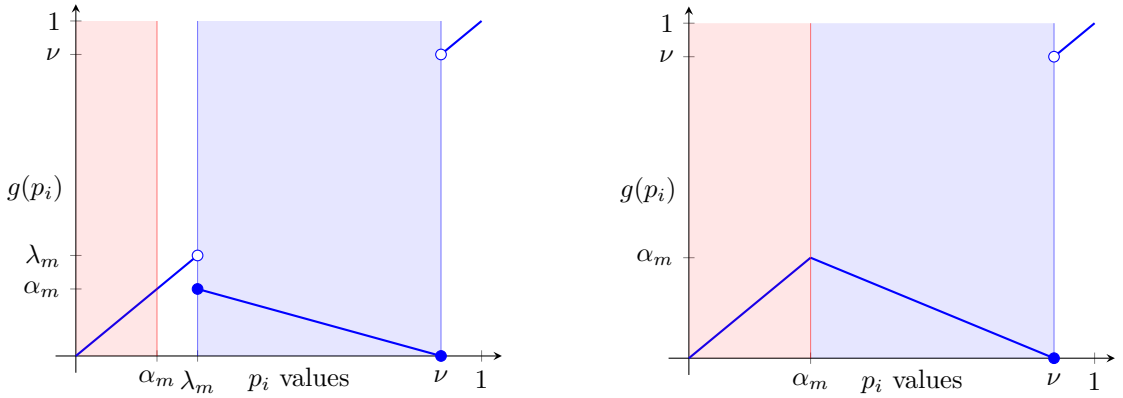
where $0 < \alpha_m \leq \lambda_m < \nu \leq 1$, as pictured in Figure 2.2.

Using this, two other variables are then defined: the auxiliary indicator $b_i := \mathbb{I}(p_i \neq m_i)$, and the sign indicator $s_i := \operatorname{sgn}(Z_i)(-1)^{b_i}$. Applying these definitions, the procedure reveals the triplet $(\boldsymbol{x}_i, m_i, s_i)$ for masked data, and $(\boldsymbol{x}_i, Z_i)$ is revealed for unmasked data. Hence, the final set of data the model receives during the $t$th iteration under this formulation is

$$\mathcal{D}_t := \{(\boldsymbol{x}_i, m_i, s_i) \,:\, i \in \mathcal{M}_t\} \cup \{(\boldsymbol{x}_i, Z_i) \,:\, i \notin \mathcal{M}_t\}.$$

We can simplify this representation with some manipulation. The revealing of $(m_i, s_i)$ to the algorithm implies two candidate values for a masked $Z_i$,

$$Z_{t,i}^{(0)} = s_i \phi^{-1}\left(1 - \frac{m_i}{2}\right), \quad Z_{t,i}^{(1)} = -s_i \phi^{-1}\left(1 - \frac{\nu - \zeta m_i}{2}\right) \tag{2.4}$$




(a) $g(p)$ with $\alpha_m < \lambda_m$

(b) $g(p)$ with $\alpha_m = \lambda_m$

Figure 2.2: Masking function $g(p)$ defined in Eq (2.3) across two different assignments of $\alpha_m$; $p$-values outside of the red or blue regions cannot be masked. Note that a given $p_i$ can only be masked if $p_i = g(p)$ has multiple feasible solutions. In (a) this occurs if $g(p_i) \leq \alpha_m$; in (b) this occurs if $g(p_i) < \alpha_m$.

where $Z_i = Z_{t,i}^{(b)}$ if and only if $b_i = b$. Given unmasked data only has one candidate value, we can write $\mathcal{D}_t \equiv \left\{(\boldsymbol{x}_i, \widetilde{Z}_{t,i}\right\}_{i=1}^{n}$ where the possible $Z$-values given the revealed data are summarised as

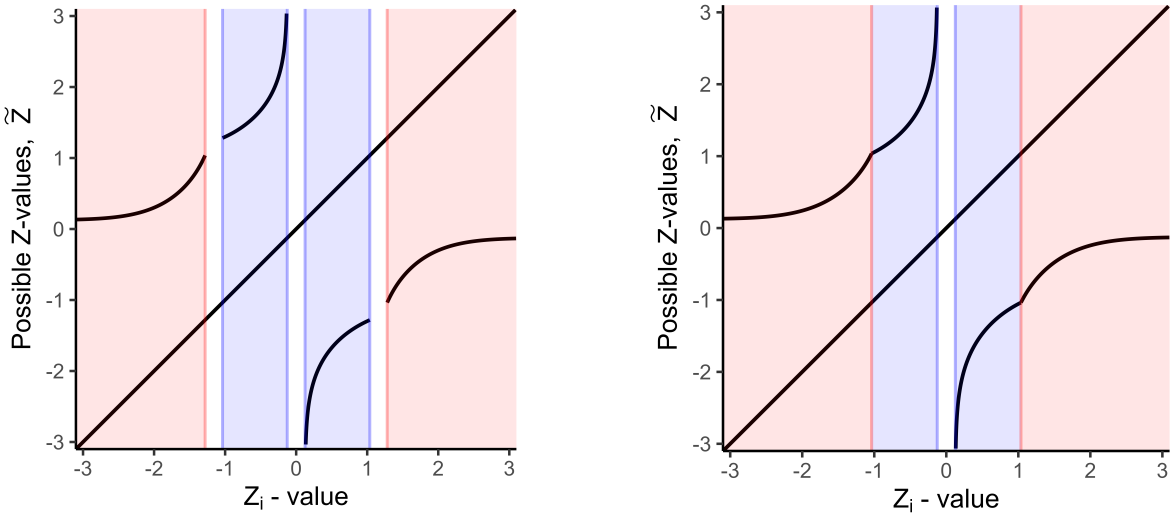
$$\widetilde{Z}_{t,i} := \begin{cases} \left\{Z_{t,i}^{(0)}, Z_{t,i}^{(1)}\right\} & i \in \mathcal{M}_t \\ Z_i & i \notin \mathcal{M}_t \end{cases} \tag{2.5}$$

A visualisation of these candidate values is provided in Figure 2.4.

As suggested from the relation in Figure 2.4, when $\alpha_m = \lambda_m$ the $Z_i$-values that can be masked are the noncentral values. We also see a large separation between the two candidate $Z_i$-values in the masking regions.

### 2.2.1. FDR Estimation

With the aforementioned masking procedure, the ZAP-GMOE procedure computes an FDP estimate in each iteration $t$ using a thresholding rule. Following the formulation in Section 2.2 in [CF21], we partition the data into acceptance and rejection regions $\mathcal{A}_t$ and $\mathcal{R}_t$ using a threshold $s_t(\boldsymbol{x}_i)$, and estimate the FDP as




(a) Masked values using $g(p)$ in Figure 2.2, (a)

(b) Masked values using $g(p)$ in Figure 2.2, (b)

Figure 2.4: A visualisation of $\widetilde{Z}_i$ as defined in Eq (2.5) over possible $Z_i$. The red and blue regions correspond directly with the matching regions in Figure 2.2 and correspond to the $Z$-values that can be masked.

$$\widehat{FDP}_t = \frac{1 + |\mathcal{A}_t|}{\zeta |\mathcal{R}_t|}$$

where

$$\mathcal{A}_t := \{i \in [n] : m_i < s_t(\boldsymbol{x}_i), \lambda_m \leq p_i \leq \nu\} \text{ and } \mathcal{R}_t := \{i \in [n] : p_i < s_t(\boldsymbol{x}_i)\}$$

In the $t$th ZAP iteration the model will only have knowledge of $|\mathcal{A}_t|$ and $|\mathcal{R}_t|$ and cannot infer the sets themselves as a result of the masking process. Moreover, for each masked $\widetilde{Z}_{t,i} = \left\{Z_{t,i}^{(0)}, Z_{t,i}^{(1)}\right\}$ as defined earlier, $Z_{t,i}^{(0)} \in \mathcal{R}_t$ and $Z_{t,i}^{(1)} \in \mathcal{A}_t$. In other words, each data falling into $\mathcal{A}_t$ or $\mathcal{R}_t$ has been masked by pairing it with a candidate from its counterpart; this reflects that these regions form a partition of the masked set, i.e. $\mathcal{M}_t = \mathcal{A}_t + \mathcal{R}_t$.

The intuition of this estimate lies in the fact that conditional on $H_{i,0}$ being true, a given $p_i$-value is uniformly distributed over $[0, 1]$. Therefore a given null $p_i$-value is $\zeta$ times more likely to land in the acceptance interval $\{p : g(p) < s_t(\boldsymbol{x}_i),\ \lambda_m \leq p_i \leq \nu\} = [\nu - \zeta s_t(\boldsymbol{x}_i), \nu]$ for $\mathcal{A}_t$ than the rejection interval $\{p : p < s_t(\boldsymbol{x}_i)\} = [0, s_t(\boldsymbol{x}_i)]$ for $\mathcal{R}_t$.

Under our assumption that the majority of the instances are null-distributed, then $|\mathcal{A}_t|/\zeta$ gives an indication of the number of masked nulls falling in $\mathcal{R}_t$, and so rejecting all values in $\mathcal{R}_t$ can be viewed as rejecting $|\mathcal{A}_t|/\zeta$ nulls.

The threshold $s_t(\boldsymbol{x}_i)$ can be freely chosen in the implementations of ZAP and AdaPT-GMM$_g$ at each step of the algorithm, provided that it is non-increasing in $t$, i.e. $s_{t+1}(\boldsymbol{x}_i) \leq s_t(\boldsymbol{x}_i)$ for all $i \in [n]$, and that the $t$th update is dependent only on the data visible to the algorithm, i.e. $(\mathcal{D}_t, |\mathcal{A}_t|, |\mathcal{R}_t|)$.

By default, the algorithm allows all masked instances to contribute to $\mathcal{A}_t$ and $\mathcal{R}_t$, i.e. we set the threshold $s_t(\boldsymbol{x}_i) = \mathbb{I}(i \in \mathcal{M}_t)$. Consequently, the sets $\mathcal{A}_t, \mathcal{R}_t$ are exactly the data that fall in the red/blue regions in Figures 2.2, 2.4.

## 2.3. Ranking

In the general sense, an optimal $\alpha$-level $Z$-value procedure is one that maximises the number of true positives while controlling the number of false positives. One implementation of an optimal or oracle procedure (see [SC07], or Appendix A [LS21]) opts to reject a given hypothesis $H_{i,0}$ if the corresponding conditional posterior probability $\mathbb{P}(\mathbb{H}_i = 0|Z_i, \boldsymbol{x}_i)$ is less than a data-determined threshold $t$. The $\mathbb{P}(\mathbb{H}_i = 0|Z_i, \boldsymbol{x}_i)$ are also called conditional local false discovery rates (Clfdr)[1]. The procedure defines

$$t = \max_{j \in [n]} \left\{ L_{(j)} \,:\, \frac{1}{j} \sum_{i=1}^{j} L_{(i)} \leq \alpha \right\}$$

where $L_{(1)} \leq \cdots \leq L_{(n)}$ are the order statistics of the Clfdr probabilities. This procedure is proven to maximise the expected number of true discoveries while controlling the FDR below the specified $\alpha$. However, in general the Clfdr probabilities are unknown, so the performance of the oracle procedure serves as an important measure to compare the performance of data-driven models against.

In[LS21], the rank-and-threshold methodology of the optimal procedure is emulated,

[1] A terminology which was introduced in [Efr+01].

but without directly attempting to estimate the CLfdr probabilities [2] This is achieved by using an *assessor function* as a proxy for the Clfdr to rank and select the most likely masked value to be null-distributed. The ZAP-GMOE procedure follows a similar strategy, using a modified assessor function discussed in Eq (6) of [CF21]. We rank masked instances by computing an estimate for $\mathbb{P}(b_i = 1 \mid \boldsymbol{x}_i, m_i, s_i)$ given by

$$\hat{q}_{t,i} := \frac{\zeta \hat{f}(Z_{t,i}^{(1)} \mid \boldsymbol{x}_i)/\phi(|Z_{t,i}^{(1)}|)}{\zeta \hat{f}(Z_{t,i}^{(1)} \mid \boldsymbol{x}_i)/\phi(|Z_{t,i}^{(1)}|) + \hat{f}(Z_{t,i}^{(0)} \mid \boldsymbol{x}_i)/\phi(|Z_{t,i}^{(0)}|)} \tag{2.6}$$

where $i \in \mathcal{M}_t$ and $\hat{f}(\cdot \mid \boldsymbol{x}_i, \hat{\boldsymbol{w}}, \hat{\boldsymbol{\beta}}, \hat{\sigma}_1^2, \ldots, \hat{\sigma}_K^2)$ is the estimated GMoE mixture distribution for $Z$ conditional on $\boldsymbol{x}$ and the information available to the algorithm. (See Appendix A.2 for the derivation of this modified form).

## 2.4. Implementation Details

The ZAP-GMOE algorithm is implemented in an R package available here.

The user should provide an $n$-length statistic vector $Z$, and an $n \times p$ covariate matrix $X$ (i.e. not including an intercept column). $Z$ may be rescaled by instance sample errors $\sigma_i$, and $X$ may be rescaled, but these transformations are not performed by the package.

Unless the user specifies otherwise, the choice of $\alpha_m, \lambda_m$, $\nu$ and $\zeta$ is made automatically following the convention in Section 3.2 of [CF21] - i.e. we assign

$$\zeta = \max\left(2, \min\left(\frac{1}{\alpha}, \frac{300}{n\alpha}\right)\right), \quad \nu = 0.9, \quad \alpha_m = \lambda_m = \frac{\nu}{\zeta + 1}$$

This assignment yields a masking function identical to Figure 2.2(b) for $n \geq 3000$, which is the case for most of the investigated datasets in Chapter 3. This assignment shares the benefits discussed in Section 2.2 of [CF21], in that setting $\nu < 1$ excludes $p$-values close to 1. Without this, mostly null-distributed datasets tend to generate many $p$-values near 1, which inflates $|\mathcal{A}_t|$ and hurts the power of the algorithm.

[2] See [CSW19] for a $Z$-value adaptive method estimating the Clfdr directly.

The EM algorithm (see 2) by default runs for `max_it` $= 50$ iterations with a convergence tolerance within $10^{-4}$; this is not always achieved but has been found empirically to be sufficient. In this paper, we use `max_it` $= 50$ for $K = 2$ and `max_it` $= 150$ for $K = 3$, though generally far less iterations are required. The EM algorithm is accelerated by using the `RcppArmadillo` package provided in [SC16; SC18] to implement low-level operations in `C++` and the `SQUAREM` package for faster iteration ([DV20]). The EM model parameters $\boldsymbol{w}$ and $\boldsymbol{\beta}$ are both initialised as zero-vectors, while the expert variances $\boldsymbol{\sigma}^2$ are set randomly.

By default, the working model uses $K = 2$ expert distributions and penalties $\lambda_k = \gamma_k = 1,\ k \in [K]$, which have been found to be robust in experimentation.

Both gating network implementations discussed in Appendix A.3.2 are provided. As both algorithms converge to the same local maximiser, we set the the faster Proximal Newton-type procedure as the default.

# Chapter 3

# Numerical Studies

The following models are executed on the simulation datasets discussed in this section:

- The finite-sample ZAP model which we have extended in this paper. Following the conventions in their paper, we choose to re-fit the model every $\lceil n/100 \rceil$ iterations, i.e. `nfits` $= 100$.
- The Asymptotic ZAP model also introduced in [LS21] using default specifications.
- The AdaPT-GMM$_g$ model in [CF21], which we also set to re-fit the model every $\lceil n/100 \rceil$ iterations.
- The ZAP-GMOE Algorithm 1 with the implementation choices discussed in 2.4. We observe the performance while varying $K \in \{2, 3\}$ and $\nu \in \{0.8, 0.9\}$ and with `nfits` $= 500$, which corresponds to four simulated methods in the results below. I refer to these with the notation ZAP-GMOE$(K, \nu)$ in the below discussion.It is important to emphasise that because of the significantly larger number number of fits, that **the below results do not fairly compare ZAP-GMOE performance to the other models**, though some insights can still be made.

# 3.1. Simulated Data

## 3.1.1. Simulation Models

We use the simulation data models specified in Section 4.1 of [LS21] which for convenience are re-stated here. We simulate the dataset $\{(Z_i, \boldsymbol{x}_i)\}_{i=1}^n$ with two-dimensional covariates ($p = 2$) and for 5000 hypotheses ($n = 5000$). First, the $\boldsymbol{x}_i$ are independently generated from $\mathrm{N}\left(\mathbf{0}, \begin{bmatrix} 1/2 & 0 \\ 0 & 1/2 \end{bmatrix}\right)$. Conditioning on $\boldsymbol{x}_i$, each $Z_i$ is generated from the normal mixture model with density

$$f_g(Z \mid \boldsymbol{x}) = (1 - w_l(\boldsymbol{x}) - w_r(\boldsymbol{x}))\phi(Z) + w_l(\boldsymbol{x})\phi\left(Z \mid \mu_l(\boldsymbol{x}), \sigma_g^2\right) + w_r(\boldsymbol{x})\phi\left(Z \mid \mu_r(\boldsymbol{x}), \sigma_g^2\right)$$

In this model, $w_l(\boldsymbol{x})$ and $w_r(\boldsymbol{x})$ are probabilities conditional on $\boldsymbol{x}$ that control the presence of alternative data, while $\mu_l(\boldsymbol{x}), \mu_r(\boldsymbol{x})$ and $\sigma_g^2 = 1$ are the mean effects and variance of the alternative Gaussian components respectively.

In the simulation studies we set $\sigma_g^2 = 1$. The remaining parameters $\left\{w_l(\boldsymbol{x}), w_r(\boldsymbol{x}), \mu_l(\boldsymbol{x}), \mu_r(\boldsymbol{x}), \sigma_g^2\right\}$ are defined across three setups outlined below. For shorthand we write the sum of the covariates as $x_\circ := \sum_{j=1}^{p} x_j$.

**Setup 1**: This generates data with a single alternative, where an increasing $x_\circ$ reduces the non-null probability $w_r(\boldsymbol{x})$ and mean effect $\mu_r(\boldsymbol{x})$. We use

$$w_r(\boldsymbol{x}) = \frac{1}{1 + \exp(-\eta - \zeta x_\circ)}, \quad w_l(\boldsymbol{x}) = 0, \quad \mu_r(\boldsymbol{x}) = \frac{2\epsilon}{1 + \exp(-\zeta x_\circ)}, \quad \mu_l(\boldsymbol{x}) = 0$$

In the simulation studies we iterate over all possible combinations of $\epsilon \in \{1.3, 1.5, 1.7, 1.9, 2.1\}$ , $\zeta \in \{0, 0.5, 1\}$ and $\eta = -2$. Across the three setups, $\eta$ can be viewed as a control for the frequency of non-null data[1]; lower $\eta$ corresponds with higher values of $1 - w_l(\boldsymbol{x}) - w_r(\boldsymbol{x})$ and hence fewer alternatives. Meanwhile, $\zeta$ controls what [LS21] describes as "*informativeness*" of the covariates; larger $\zeta$ allows $x_\circ$ to more strongly influence the latent mixing weights and/or mean effects.

[1]The probability of being non-null is $w_l(\boldsymbol{x}, \eta) + w_r(\boldsymbol{x}, \eta)$, which attains a minimum when $\zeta$=0. Setup 1, 3 have a base signal of 12%, and Setup 2 14%.

**Setup 2**: In this setup two alternative components are used, which have fixed but opposing mean effects, and the degree of imbalance between their weights is dependent on $x_\circ$. We use

$$w_r(\boldsymbol{x}) = \frac{\exp(\zeta x_\circ)}{\exp(-\eta) + \exp(-\zeta x_\circ) + \exp(\zeta x_\circ)}, \quad w_l(\boldsymbol{x}) = \frac{\exp(-\zeta x_\circ)}{\exp(-\eta) + \exp(-\zeta x_\circ) + \exp(\zeta x_\circ)}$$

$$\mu_r(\boldsymbol{x}) = \epsilon, \quad \mu_l(\boldsymbol{x}) = -\epsilon$$

In the simulation studies we iterate over all possible combinations of
$\epsilon \in \{1.3, 1.5, 1.7, 1.9, 2.1\}$, $\zeta \in \{0, 0.7, 1\}$ and $\eta = -2.5$.

**Setup 3**: In this setup, larger values of $x_\circ$ increase the disparity between the alternative means. We use

$$w_r(\boldsymbol{x}) = w_l(\boldsymbol{x}) = \frac{1/2}{1 + \exp(-\eta)}, \quad \mu_r(\boldsymbol{x}) = \frac{2\epsilon}{1 + \exp(-\zeta x_\circ)}, \quad \mu_l(\boldsymbol{x}) = \frac{-2\epsilon}{1 + \exp(\zeta x_\circ)}$$

In the simulation studies we iterate over all possible combinations of
$\epsilon \in \{1.3, 1.5, 1.7, 1.9, 2.1\}$, $\zeta \in \{0, 1.5, 3\}$ and $\eta = -2$.

### 3.1.2. Results

We present the results of two simulations below in Figure 3.1 and Figure 3.2, which are reproductions of the results established in [LS21]. We apply the list of methods above with the FDR target $\alpha = 0.05$, and simulate all three setups on their parameter combinations described above. 150 random instances are generated for each combination, totalling 6,750 datasets.

**Figure 3.1 (Baseline)**: The ZAP-GMOE configurations control the FDR at the desired level of 0.05, except in Setup 2, $\epsilon = 1.9$. Since the model controls FDR in expectation, and given the general consistency in which ZAP-GMOE$(2, 0.9)$ respects the FDR, this is likely resolved by increasing the number of repetitions. Asymptotic ZAP violates the FDR more frequently, but its algorithm is only expected to control the FDR asymptotically.

Focusing on the TPR, there is a consistent under-performance of ZAP-GMOE in Setups

1 and 2 in all four configurations compared to the ZAP and AdaPT-GMM$_g$ models, with ZAP-GMOE$(3, 0.8)$ being generally under-powered compared to the other configurations. In Setup 3 however, we observe similar performance in ZAP-GMOE$(2, 0.8)$ (0.328) and AdaPT-GMM$_g$ (0.336), though this is still under-powered compared to finite-sample ZAP (0.347). Curiously, there is no consistent ranking between the four ZAP-GMOE configurations in terms of TPR across the three setups. It is worth noting that Asymptotic ZAP consistently performs as well or better than the remaining models, albeit without any theoretical guarantee for finite-sample FDR control.

**Figure 3.2 (Smaller Baseline)**: Similar to the larger baseline simulation, all the finite-sample models respect the nominal FDR with the exception of ZAP-GMOE$(2, 0.8)$ in Setup 3, $\epsilon = 2.1$, but this is also expected to resolve with more repetitions.
There are several interesting observations here. Firstly, the $K = 3$ models are consistently under-powered in this setting relative to the other models, which is most likely due to their higher model complexity underfitting the smaller datasets. Secondly, the $K = 2$ models consistently overtake finite-sample ZAP in Setup 3, though given `nfits` is not common across these models, this is likely not the case in general. The significant performance gap in Setup 1 in Figure 3.1 is repeated here, though $K = 2$ has performed better in this setting.

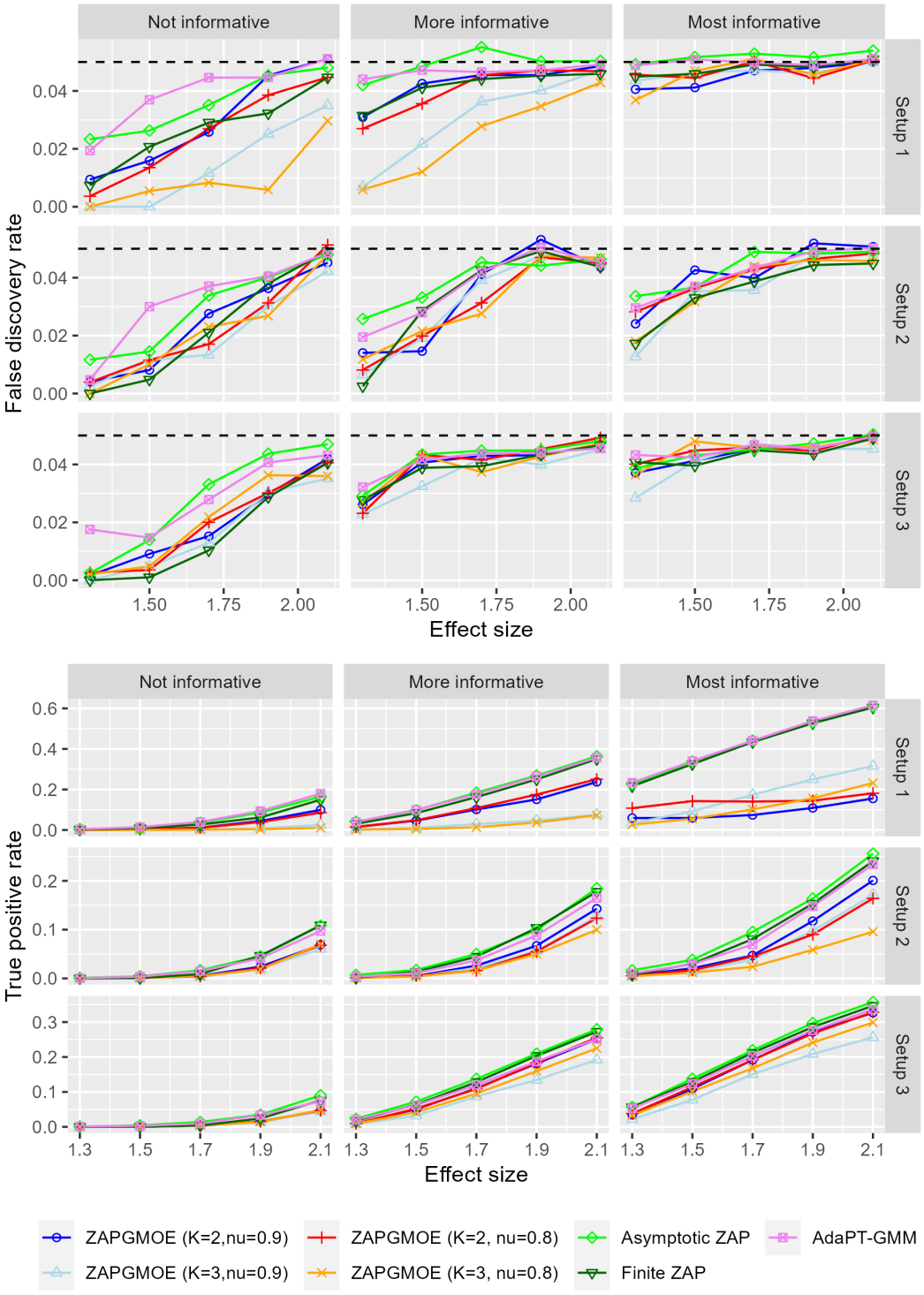



Figure 3.1: **"Baseline"**: The FDR and TPR for the methods and setups introduced above, with $n = 5000$ and on 150 repetitions for each parameter combination. The three $\zeta$ values per setup correspond to "Not Informative", "More Informative" and "Most Informative" while $\epsilon$ ("Effect Size") is the $x$-axes of each plot. The desired $\alpha = 0.05$ is indicated in the black horizontal lines.

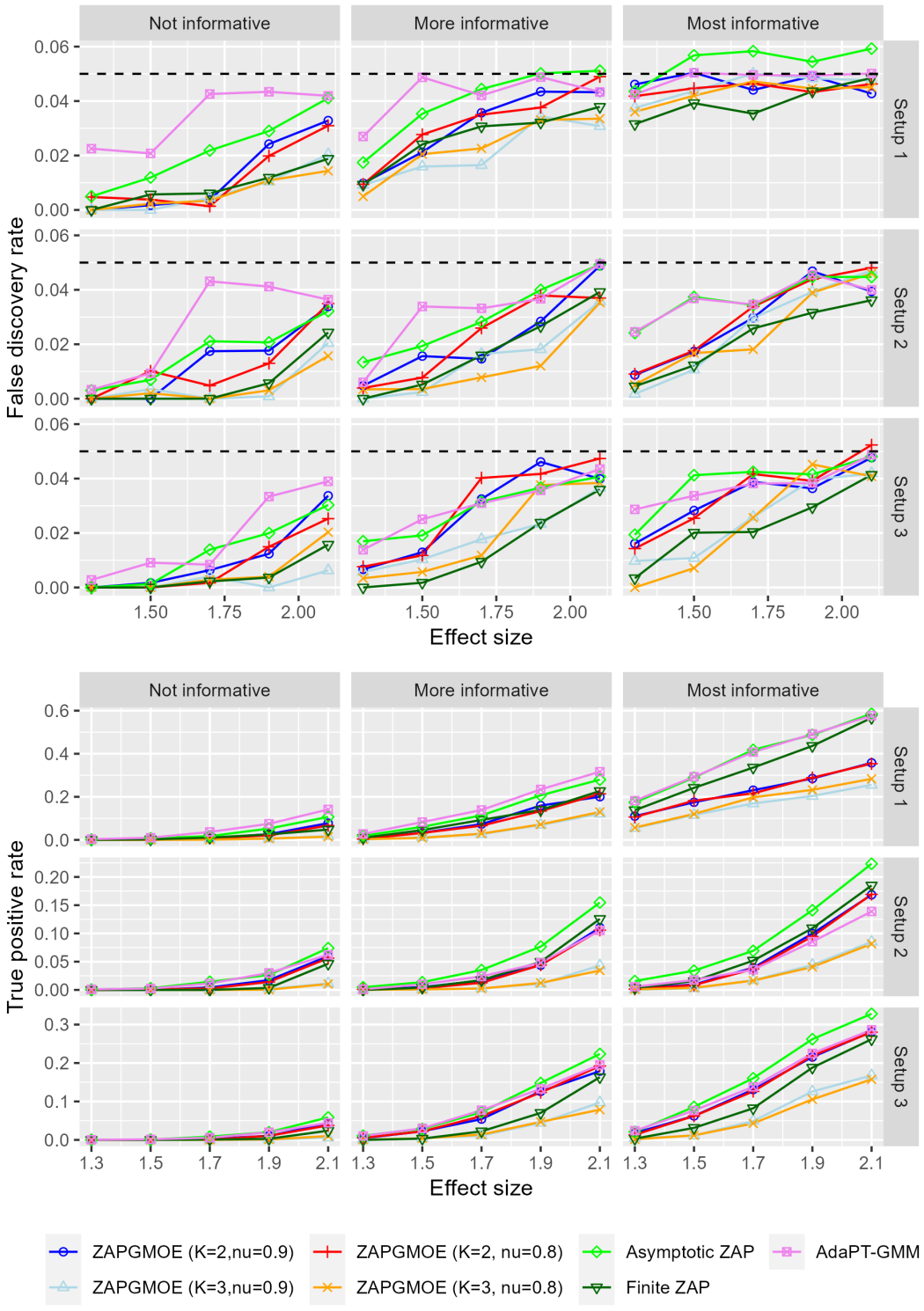



Figure 3.2: **"Smaller Baseline"**: The results of the simulation procedure in Figure 3.1 but using $n = 1000$.

# Chapter 4

# Discussion

The four simulated ZAP-GMOE configurations observed were chosen to evaluate the effect of varying the two hyper-parameters $K$ and $\nu$ on model performance. The complexity of the GMoE model is governed by the number of components $K$, and the proportion of the dataset that is masked is controlled by $\nu$, as it sets the upper limit for $p$-values that can be masked (as shown in Figure 2.2)).

Amongst the analysed ZAP-GMOE model configurations, the best performance was in Setup 3 with $n = 5000$. This spike in performance is also consistent with the observation in Figure B.1 that additional covariate models also exhibited strong performances in Setup 3. The stronger performance here may be due to the unbalanced effect of covariates on the component means and weights in Setup 3 - the covariate arguably has the strongest effect on the mixture density of the three setups, allowing for stronger non-null signals. Contrastingly, the most significant under-performance was in Setup 1 for $n = 5000$, where the most informative $\zeta$ corresponded with a poorer performance.

There is evidence to suggest that the model can be improved with a model selection or hyper-parameter tuning approach. The ranking in performance of the ZAP-GMOE configurations in the $n = 1000$ setting were identical in $n = 5000$ for Setup 2 and 3, and tuning of the GMoE regularisation vectors $\lambda$ and $\gamma$ was only investigated briefly to

select the default values of 1. A model selection approach might partition the dataset into smaller strata, and the performance on separate parameter choices evaluated to identify an optimal combination of $K, \nu$ and GMoE penalties $\lambda, \gamma$ - however, this may violate the FDR control guarantee. An approach like in [CF21] may be more suitable - where the authors seek a combination of parameters just to maximise the fit of the underlying data model prior to beginning the adaptive masking algorithm.

The use of the regularised GMoE model in [HC19] was motivated by a desire to fit sparser, higher-dimensional feature spaces with high levels of information on the $Z$-values. The simulations executed in this paper involve 2-dimensional, dense covariate distributions, and so we have not explored this motivation. Aside from the general under-performance of the model even with higher `nfits`, whether this model is suitable for the motivating data distributions is currently still unaddressed.

This report also excludes any application of the ZAP-GMOE model to real data, such as the various RNA-Seq datasets testing for differential gene expression explored in [LS21; CF21], which are both higher-dimensional and sparser than the simulations considered here. The breadth of simulations can also be extended; future investigations should, aside from using the same `nfits` for ZAP-GMOE, could also explore variations in the data-generating variance $\sigma^2$, and in the proportion of non-nulls (controlled by $\eta$), which would give more evidence to the robustness of the algorithm.

One other limitation of the ZAP-GMOE model is that it is currently limited to testing the point null hypotheses $H_{i,0} : \mu_i = 0$, but should be extensible with modification to the masking procedure to handle interval or one-sided testing, in a similar manner to the suggestions in the Appendix of [CF21]. This may benefit from a more appropriate underlying data model to better capture the symmetry or asymmetry in these scenarios.

Overall, the general observation from 3.1 and B.1 is that even with a generous value for `nfits`, the ZAP-GMOE model in its current form is under-powered compared to the ZAP and AdaPT-GMM$_g$ models, which is a clear point of future investigation. A more extensive simulation procedure involving sparser and higher-dimensional covariate

data and a preliminary model selection step will offer more conclusive insight into the feasibility of the regularised MGoE model in a $Z$-value covariate-adaptive procedural context.

# References


[Jac+91] Robert A. Jacobs et al. "Adaptive Mixtures of Local Experts". In: *Neural Computation* 3 (1991), pp. 79–87.

[BH95] Yoav Benjamini and Yosef Hochberg. "Controlling the false discovery rate: a practical and powerful approach to multiple testing". In: *Journal of the royal statistical society series b-methodological* 57 (1995), pp. 289–300.

[Efr+01] Bradley Efron et al. "Empirical Bayes Analysis of a Microarray Experiment". In: *Journal of the American Statistical Association* 96.456 (2001), pp. 1151–1160.

[MP04] G.J. McLachlan and D. Peel. *Finite Mixture Models*. Wiley Series in Probability and Statistics. Wiley, 2004.

[SC07] Wenguang Sun and T. Tony Cai. "Oracle and Adaptive Compound Decision Rules for False Discovery Rate Control". In: *Journal of the American Statistical Association* 102 (2007), pp. 901–912.

[YWG12] Seniha Esen Yuksel, Joseph N. Wilson, and Paul D. Gader. "Twenty Years of Mixture of Experts". In: *IEEE Transactions on Neural Networks and Learning Systems* 23.8 (2012), pp. 1177–1193.

[HTW15] T. Hastie, R. Tibshirani, and M. Wainwright. *Statistical Learning with Sparsity: The Lasso and Generalizations*. Chapman & Hall/CRC Monographs on Statistics & Applied Probability. CRC Press, 2015.

[Sco+15] James G. Scott et al. “False Discovery Rate Regression: An Application to Neural Synchrony Detection in Primary Visual Cortex”. In: *Journal of the American Statistical Association* 110.510 (2015), pp. 459–471.

[BC16] Rina Barber and Emmanuel Candes. “A knockoff filter for high-dimensional selective inference”. In: *The Annals of Statistics* 47 (Feb. 2016).

[Ign+16] Nikolaos Ignatiadis et al. “Data-driven hypothesis weighting increases detection power in genome-scale multiple testing”. In: *Nature methods* 13.7 (May 2016), pp. 577–580.

[LF16] Lihua Lei and William Fithian. “AdaPT: An Interactive Procedure for Multiple Testing with Side Information”. In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 80 (Sept. 2016).

[NLM16] Hien D. Nguyen, Luke R. Lloyd-Jones, and Geoffrey J. McLachlan. “A Universal Approximation Theorem for Mixture-of-Experts Models”. In: *Neural Computation* 28.12 (Dec. 2016), pp. 2585–2593.

[SC16] Conrad Sanderson and Ryan R. Curtin. “Armadillo: a template-based C++ library for linear algebra”. In: *Journal of Open Source Software* 1.2 (2016), p. 26.

[SC18] Conrad Sanderson and Ryan Curtin. “A User-Friendly Hybrid Sparse Matrix Class in C++”. In: *Lecture Notes in Computer Science (LNCS)* 10931 (2018), pp. 422–430.

[CSW19] T Tony Cai, Wenguang Sun, and Weinan Wang. “Covariate-assisted ranking and screening for large-scale two-sample inference (with discussion)”. In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 87 (2019), pp. 187–217.

[Har+19] Rayna M. Harris et al. “Hippocampal transcriptomic responses to enzyme-mediated cellular dissociation”. In: *Hippocampus* 29.9 (2019), pp. 876–882.

[HC19] Bao Tuyen Huynh and Faicel Chamroukhi. "Estimation and Feature Selection in Mixtures of Generalized Linear Experts Models". In: *arXiv* (July 2019).

[DV20] Yu Du and Ravi Varadhan. "SQUAREM: An R Package for Off-the-Shelf Acceleration of EM, MM and Other EM-Like Monotone Algorithms". In: *Journal of Statistical Software* 92.7 (2020), pp. 1–41.

[Yur+20] Ronald Yurko et al. "A selective inference approach for false discovery rate control using multiomics covariates yields insights into disease risk". In: *Proceedings of the National Academy of Sciences* 117.26 (2020), pp. 15028–15035.

[ZC20] Xianyang Zhang and Jun Chen. "Covariate Adaptive False Discovery Rate Control With Applications to Omics-Wide Multiple Testing". In: *Journal of the American Statistical Association* (July 2020), pp. 1–31.

[CF21] Patrick Chao and William Fithian. "AdaPT-GMM: Powerful and robust covariate-assisted multiple testing". In: *arXiv* (2021). URL: `https://arxiv.org/abs/2106.15812`.

[LS21] Dennis Leung and Wenguang Sun. "ZAP: $Z$-value adaptive procedures for false discovery rate control with side information". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 84 (2021), pp. 1886–1946.

# Appendix A

# Theoretical Results

## A.1. Proof for finite-sample FDR control

The proof for Theorem 2.1 is mostly similar to the proof in Appendix A.1 in [CF21]. Following their structure:

$$
\begin{aligned}
\mathcal{C}_t &:= \mathcal{H}_0 \cap \mathcal{M}_t && \text{[Masked null values]} \\
b_i &:= \mathbb{I}\left(\lambda_m \le p_i \le \nu\right) && \text{[Domain indicator]} \\
\mathbb{U}_t &:= \sum_{i \in \mathcal{C}_t} b_i \\
\mathbb{V}_t &:= \sum_{i \in \mathcal{C}_t} (1 - b_i) \equiv |\mathcal{C}_t| - \mathbb{U}_t \\
\mathcal{F}_{-1} &:= \sigma\left((\boldsymbol{x}_i, m_i, s_i)_{i\in[n]}, \mathcal{M}_0, \{b_i : i \in \mathcal{M}_0^c\}\right) \\
\mathcal{F}_t &:= \sigma\left((\boldsymbol{x}_i, m_i, s_i)_{i\in[n]}, |\mathcal{A}_t|, |\mathcal{R}_t|, \mathcal{M}_t, \{b_i : i \in \mathcal{M}_t^c\}\right) \\
\mathcal{G}_{-1} &:= \sigma\left((\boldsymbol{x}_i, m_i, s_i)_{i\in[n]}, (b_i)_{i\notin\mathcal{H}_0}\right) \\
\mathcal{G}_t &:= \sigma\left(\mathcal{G}_{-1}, \mathcal{C}_t, (b_i)_{i\notin\mathcal{C}_t}, \mathbb{U}_t\right) \equiv \sigma\left((\boldsymbol{x}_i, m_i, s_i)_{i\in[n]}, (b_i)_{i\notin\mathcal{C}_t}, \mathcal{C}_t, \mathbb{U}_t\right) && \text{(A.1)}
\end{aligned}
$$

Also recall that Eq (2.4) and Eq (2.5) is a function of the data tuple $(m_i, s_i)$; more generally, given the tuple $(m_i, s_i, b_i)$ one can infer $\widetilde{Z}_{t,i}$.

To assist in the proof, I also restate these two key lemmas:

**Lemma A.1 (cf. Lemma A.1, [CF21]).** *Assume that the null p-values are mutually*

*independent, and independent of the non-null p-values, and further that the null p-values have non-decreasing density (same as in Theorem 2.1). Then*

$$\mathbb{P}(b_i = 1 \,|\, \mathcal{G}_{-1}) \geq \frac{\zeta}{1+\zeta} \quad \forall\, i \in \mathcal{H}_0$$

**Lemma A.2 (cf. Lemma 2, [LF16]).** *Suppose $\{b_i\}_{i=1}^n$ are i.i.d. Bernoulli random variables conditional on the $\sigma$-algebra $\mathcal{G}_{-1}$, with $\mathbb{P}(b_i = 1 \,|\, \mathcal{G}_{-1}) = \rho_i \geq \rho > 0$ a.s., and suppose that $[n] \supseteq \mathcal{C}_0 \supseteq \mathcal{C}_1 \supseteq \ldots$, where each $\mathcal{C}_{t+1}$ is $\mathcal{G}_t$-measurable, with*

$$\mathcal{G}_t = \sigma\left(\mathcal{G}_{-1}, \mathcal{C}_t, (b_i)_{i \notin \mathcal{C}_t}, \sum_{i \in \mathcal{C}_t} b_i\right).$$

*Then if $\hat{t}$ is a finite stopping time (a.s.) with respect to the filtration $(\mathcal{G}_t)_{t \geq 0}$ then*

$$\mathbb{E}\left[\frac{1+|\mathcal{C}_{\hat{t}}|}{1+\sum_{i \in \mathcal{C}_{\hat{t}}} b_i} \,|\, \mathcal{G}_{-1}\right] \leq \rho^{-1}$$

Now we can proceed with the proof.

*Proof of Theorem 2.1.* Define $\hat{t}$ as the stopping time for the ZAP-GMOE procedure. Then by definition

$$\widehat{\mathrm{FDP}}_{\hat{t}} = \frac{1+|\mathcal{A}_{\hat{t}}|}{\zeta|\mathcal{R}_{\hat{t}}|} \leq \alpha \tag{A.2}$$

By simple definitions, following the idea in Eq (16) in [CF21] we have

$$\begin{aligned}
\mathrm{FDP}_{\hat{t}} &= \frac{\mathbb{V}_{\hat{t}}}{|\mathcal{R}_{\hat{t}}| \vee 1} = \frac{\mathbb{V}_{\hat{t}}}{(1+\mathbb{U}_{\hat{t}})/\zeta} \frac{(1+\mathbb{U}_{\hat{t}})/\zeta}{|\mathcal{R}_{\hat{t}}| \vee 1} \\
&\leq \frac{\mathbb{V}_{\hat{t}}}{(1+\mathbb{U}_{\hat{t}})/\zeta} \frac{(1+|\mathcal{A}_{\hat{t}}|)/\zeta}{|\mathcal{R}_{\hat{t}}| \vee 1} && \left[\text{As } \mathbb{U}_t \leq \sum_{i \in \mathcal{M}_t} b_i \leq |\mathcal{A}_t|\right] \\
&\leq \frac{\zeta \mathbb{V}_{\hat{t}}}{1+\mathbb{U}_{\hat{t}}} \times \left(\widehat{\mathrm{FDP}}_{\hat{t}}\right) = \alpha\zeta \frac{\mathbb{V}_{\hat{t}}}{1+\mathbb{U}_{\hat{t}}} && [\text{See Eq (A.2)}]
\end{aligned} \tag{A.3}$$

Applying the theorem assumptions, by Lemma A.1 we find for $i \in \mathcal{H}_0$ that $\rho_i := \mathbb{P}(b_i = 1 \,|\, \mathcal{G}_{-1}) \geq \frac{\zeta}{1+\zeta}$.

We also establish that $\mathcal{F}_t \subseteq \mathcal{G}_t$ by demonstrating that each component used in generating $\mathcal{F}_t$ is $\mathcal{G}_t$-measurable.

By definition, the data tuples are $\mathcal{G}_t$-measurable as $(\boldsymbol{x}_i, s_i, m_i) \in_m \mathcal{G}_{-1} \subseteq \mathcal{G}_t$. Secondly, $\{b_i : i \in \mathcal{M}_t^c\} \subseteq \{b_i : i \notin \mathcal{C}_t\}$ and $(b_i)_{i \notin \mathcal{C}_t} \in_m \mathcal{G}_t$ imply that $\{b_i : i \in \mathcal{M}_t^c\} \in_m \mathcal{M}_t$.

Now note that $|\mathcal{A}|_t = \mathbb{U}_t + |\{i \notin \mathcal{H}_0 \,:\, b_i = 1 \text{ and } i \in \mathcal{M}_t\}|$. Since $\mathbb{U}_t \in_m \mathcal{G}_t$ by construction, and $\{b_i : i \in \mathcal{M}_t \setminus \mathcal{H}_0\} \subseteq \{b_i : i \notin \mathcal{C}_t\} \in_m \mathcal{G}_t$, then $|\mathcal{A}_t|$ is a function of $\mathcal{G}_t$-measurable values and so must also be measurable.

Similarly, note that $|\mathcal{R}|_t = \mathbb{V}_t + |\{i \notin \mathcal{H}_0 \,:\, 0 \leq p_i \leq \alpha_m \text{ and } i \in \mathcal{M}_t\}|$. Given that $b_i = 0$ implies $0 \leq p_i \leq \alpha_m$ for any $i \in \mathcal{M}_t$, then $|\mathcal{R}_t|$ is a function of the $\mathcal{G}_t$-measurable $\mathbb{V}_t$ and $\{b_i : i \in \mathcal{M}_t \setminus \mathcal{H}_0\}$ and therefore must also be $\mathcal{G}_t$-measurable.

Lastly, $\{b_i : i \in \mathcal{M}_t^c\} \in_m \mathcal{G}_t$ implies $\mathcal{M}_t^c \in_m \mathcal{G}_t$, and so by closure under complements $\mathcal{M}_t \in_m \mathcal{G}_t$.

As a result, $\hat{t}$ is now a stopping time with respect to $\{\mathcal{G}_t\}_{t \geq 0}$. We now apply Lemma A.2 to the conditional expectation of the result in (A.3):

$$\begin{aligned}
\mathbb{E}\left[\mathrm{FDP}_{\hat{t}} \,|\, \mathcal{G}_{-1}\right] &\leq \alpha\zeta\mathbb{E}\left[\frac{\mathbb{V}_{\hat{t}}}{1+\mathbb{U}_{\hat{t}}} \,|\, \mathcal{G}_{-1}\right] \\
&= \alpha\zeta\mathbb{E}\left[\frac{1+|\mathcal{C}_{\hat{t}}|}{1+\mathbb{U}_{\hat{t}}} - 1 \,|\, \mathcal{G}_{-1}\right] && \text{[Definition of } \mathbb{V}_{\hat{t}}\text{]} \\
&\leq \alpha\zeta\left(\left(\frac{\zeta}{1+\zeta}\right)^{-1} - 1\right) = \alpha && \text{[Lemma A.2]}
\end{aligned}$$

By Law of Iterated Expectation we deduce

$$\mathrm{FDR} = \mathbb{E}\left[\mathbb{E}\left[\mathrm{FDP}_{\hat{t}} \,|\, \mathcal{G}_{-1}\right]\right] = \mathbb{E}\left[\mathrm{FDP}\right] \leq \alpha$$

as required. □

## A.2. Derivations for Thresholding Procedure

The estimate $\hat{q}_{t,i}$ of $q_i$ for $i \in \mathcal{M}_t$ in (2.6) can be derived by first applying Bayes' Theorem:

$$\begin{aligned} q_i &= \mathbb{P}(b_i = 1 \,|\, m_i, s_i, \boldsymbol{x}_i) \\ &= \frac{\sum_{k=1}^{K} \mathbb{P}(b_i = 1, m_i, s_i \,|\, \Gamma_i = k, \boldsymbol{x}_i)\, \mathbb{P}(\Gamma_i = k \,|\, \boldsymbol{x}_i)}{\sum_{k=1}^{K} \mathbb{P}(\Gamma_i = k \,|\, \boldsymbol{x}_i)\, \{\mathbb{P}(b_i = 1, m_i, s_i \,|\, \Gamma_i = k, \boldsymbol{x}_i) + \mathbb{P}(b_i = 0, m_i, s_i \,|\, \Gamma_i = k, \boldsymbol{x}_i)\}} \end{aligned}$$

We can estimate the $\mathbb{P}(b_i = b, m_i, s_i \,|\, \Gamma_i = k, \boldsymbol{x}_i)$ terms using the result of Eq (22) from Appendix C in [CF21]. Substituting our notation, their simplification results in

$$\mathbb{P}(b_i = b, m_i, s_i \,|\, \Gamma_i = k, \boldsymbol{x}_i) \propto \frac{f_k(Z_{t,i}^{(b)} \,|\, \boldsymbol{x}_i)\zeta^b}{2\phi(|Z_{t,i}^{(b)}|)}$$

where $f_k(\cdot \,|\, \boldsymbol{x}) := f(\cdot \,|\, \boldsymbol{x}, \Gamma = k)$ is the $k$th conditional expert distribution defined in Eq (2.1).

Using the fitted mixing probabilities $\hat{\pi}_{i,k}(\hat{\boldsymbol{w}})$ to estimate the gating proportions $\mathbb{P}(\Gamma_i = k \,|\, \boldsymbol{x}_i)$, and applying the fitted conditional densities $\hat{f}_k(\cdot \,|\, \hat{\beta}_{k,0}, \hat{\boldsymbol{\beta}}_k, \hat{\sigma}_k^2)$, we can construct the estimate

$$\begin{aligned} \hat{q}_{t,i} &= \frac{\sum_{k=1}^{K} \hat{\pi}_{i,k} \hat{f}_k(Z_{t,i}^{(1)} \,|\, \boldsymbol{x}_i)\zeta/(2\phi(|Z_{t,i}^{(1)}|))}{\sum_{k=1}^{K} \hat{\pi}_{i,k} \left\{ \hat{f}_k(Z_{t,i}^{(1)} \,|\, \boldsymbol{x}_i)\zeta/(2\phi(|Z_{t,i}^{(1)}|)) + \hat{f}_k(Z_{t,i}^{(0)} \,|\, \boldsymbol{x}_i)/(2\phi(|Z_{t,i}^{(0)}|)) \right\}} \\ &= \frac{\zeta \hat{f}(Z_{t,i}^{(1)} \,|\, \boldsymbol{x}_i)/\phi(|Z_{t,i}^{(1)}|)}{\zeta \hat{f}(Z_{t,i}^{(1)} \,|\, \boldsymbol{x}_i)/\phi(|Z_{t,i}^{(1)}|) + \hat{f}(Z_{t,i}^{(0)} \,|\, \boldsymbol{x}_i)/\phi(|Z_{t,i}^{(0)}|)} \end{aligned}$$

as required, where $\hat{f}(\cdot \,|\, \boldsymbol{x}, \hat{\boldsymbol{w}}, \hat{\boldsymbol{\beta}}, \hat{\sigma}_1^2, \ldots, \hat{\sigma}_K^2)$ is the estimated GMoE mixture distribution for $Z$ conditional on $\boldsymbol{x}$ and the information available to the algorithm.

## A.3. EM Algorithm Implementation

To maximise Eq (1.8) this EM algorithm aims to maximise the penalised complete-data log-likelihood

$$\log PL_c(\boldsymbol{\theta}) = \log L_c(\boldsymbol{\theta}) - \sum_{k=1}^{K} \lambda_k \|\beta_k\|_1 - \sum_{k=1}^{K-1} \gamma_k \|\boldsymbol{w}_k\|_1 \tag{A.4}$$

where

$$\log L_c(\boldsymbol{\theta}) := \log L_c(Z_i, \Gamma_i \,|\, \boldsymbol{\theta}) = \log \left\{ \prod_{i=1}^{n} \mathbb{P}(Z_i, \Gamma_i \,|\, \boldsymbol{\theta}) \right\}$$

is the complete-data log-likelihood and $\{\gamma_k\}_{k\in[K-1]}$, $\{\lambda_k\}_{k\in[K]}$ are the lasso penalties in Eq (1.8).

The overall structure of the algorithm is provided in Algorithm 2 below, with specific update rules in subsections A.3.1 and A.3.2. In the $q$th iteration of the EM algorithm in the $t$th outer iteration of the ZAP-GMOE algorithm, we make two block updates, updating the gating and expert coefficients and intercept estimates $\hat{\boldsymbol{w}}_q, \hat{\boldsymbol{\beta}}_q$ in one block and the expert variances $\hat{\boldsymbol{\sigma}}_q^2$ in another. This is repeated until the log-likelihood achieves local convergence within some tolerance, or a pre-specified iteration limit `max_it` is reached.

### A.3.1. E-step

For the E-step at the $q$th EM iteration during the $t$th ZAP iteration, we take a conditional expectation $\mathbb{E}[\cdot \,|\, \mathcal{D}_{t,i}]$ of Eq (1.8). Defining the indicator $\Delta_{i,k} = \mathbb{I}(\Gamma_i = k)$, this step amounts to computing the estimates

$$\begin{aligned}
D^{(0)}_{t,q,i,k} &:= \mathbb{E}_{(\hat{\boldsymbol{\theta}}_{t,q})}[\Delta_{i,k} \,|\, \mathcal{D}_{t,i}] \\
D^{(1)}_{t,q,i,k} &:= \mathbb{E}_{(\hat{\boldsymbol{\theta}}_{t,q})}[\Delta_{i,k} Z_i \,|\, \mathcal{D}_{t,i}] \\
D^{(2)}_{t,q,i,k} &:= \mathbb{E}_{(\hat{\boldsymbol{\theta}}_{t,q})}\left[\Delta_{i,k} Z_i^2 \,|\, \mathcal{D}_{t,i}\right]
\end{aligned}$$

These estimates are calculated with the GMoE model in Eq (2.1) using the estimated pa-

**Algorithm 2:** EM Algorithm for ZAP-GMOE

**Data:** $\mathcal{D}_t = ((\boldsymbol{x}_i, m_i, s_i)_{i\in\mathcal{M}_t}, (\boldsymbol{x}_i, Z_i)_{i\notin\mathcal{M}_t})$

**Model Hyper-parameters:** $K \in \{2, \dots\}, \{\lambda_k\}_{k\in[K]}, \{\gamma_k\}_{k\in[K-1]}$

**Inputs:** Initial guess $\hat{\boldsymbol{\theta}}_0 = (\boldsymbol{w}_0, \boldsymbol{\beta}_0, \boldsymbol{\sigma}_0^2)$

**for** $q = 0, 1, \dots,$ `max_it` **do**

$D = (D^{(0)}, D^{(1)}, D^{(2)}) \leftarrow$ `E-step`$(\mathcal{D}_t, \hat{\boldsymbol{\theta}}_q)$; `// See A.3.1`

$\hat{\boldsymbol{w}}_{q+1} \leftarrow$ `w-update`$(\mathcal{D}_t, D^{(0)}, \hat{\boldsymbol{w}}_q)$; `// See A.3.2`

$\hat{\boldsymbol{\beta}}_{q+1} \leftarrow$ `beta-update`$(\mathcal{D}_t, D, \hat{\boldsymbol{\beta}}_q, \hat{\boldsymbol{\sigma}}_q^2)$; `// See A.3.2`

`// Second block`

$D \leftarrow$ `E-step`$(\mathcal{D}_t, \hat{\boldsymbol{\theta}}_q)$

$\hat{\boldsymbol{\sigma}}_{q+1}^2 \leftarrow$ `variance-update`$(\mathcal{D}_t, D, \hat{\boldsymbol{\beta}}_{q+1})$; `// See A.3.2`

**end for**

rameters $\hat{\boldsymbol{\theta}}_{t,q}$; however, the exact computation varies per instance depending on whether $i \in \mathcal{M}_t$.

**In the unmasked case:** Then $i \notin \mathcal{M}_t$ and so $\widetilde{Z}_{t,i} = Z_i$. Here, we estimate $D_{t,q,i,k}^{(d)}$ by using the corresponding unmasked estimates $D_{t,q,i,k}^{(d,U)}$. Firstly,

$$D_{t,q,i,k}^{(0,U)} = \mathbb{E}_{(\hat{\boldsymbol{\theta}}_{t,q})}\left[\Delta_{i,k} \,|\, \mathcal{D}_{t,i}, i \notin \mathcal{M}_t\right] = \mathbb{P}\Big(\Delta_{i,k} = 1 \,|\, Z_i, \boldsymbol{x}_i, \hat{\boldsymbol{\theta}}_{t,q}\Big) = \frac{f_k(Z_i \,|\, \boldsymbol{x}_i, \hat{\boldsymbol{\theta}}_{t,q})\pi_k(\boldsymbol{x}_i, \hat{\boldsymbol{\theta}}_{t,q})}{\sum_{k'=1}^{K} f_{k'}(Z_i \,|\, \boldsymbol{x}_i, \hat{\boldsymbol{\theta}}_{t,q})\pi_{k'}(\boldsymbol{x}_i, \hat{\boldsymbol{\theta}}_{t,q})}$$

Using this, we can express the other unmasked estimates by conditional expectation laws as

$$D_{t,q,i,k}^{(1,U)} = \mathbb{E}_{(\hat{\boldsymbol{\theta}}_{t,q})}\left[\Delta_{i,k} Z_i \,|\, \mathcal{D}_{t,i}, i \notin \mathcal{M}_t\right] = Z_i \mathbb{E}_{(\hat{\boldsymbol{\theta}}_{t,q})}\left[\Delta_{i,k} \,|\, \mathcal{D}_{t,i}\right] = Z_i D_{t,q,i,k}^{(0,U)}$$
$$D_{t,q,i,k}^{(2,U)} = \mathbb{E}_{(\hat{\boldsymbol{\theta}}_{t,q})}\left[\Delta_{i,k} Z_i^2 \,|\, \mathcal{D}_{t,i}, i \notin \mathcal{M}_t\right] = Z_i^2 \mathbb{E}_{(\hat{\boldsymbol{\theta}}_{t,q})}\left[\Delta_{i,k} \,|\, \mathcal{D}_{t,i}\right] = Z_i^2 D_{t,q,i,k}^{(0,U)}$$

**In the masked case:** then $i \in \mathcal{M}_t$ and so $\widetilde{Z}_{t,i} = \left\{Z_{t,i}^{(0)}, Z_{t,i}^{(1)}\right\} \equiv \left\{Z_i, \check{Z}_i\right\}$, where we use $\check{Z}_i = Z_{t,i}^{(1)} \mathbb{I}\left(Z_i = Z_{t,i}^{(0)}\right) + Z_{t,i}^{(0)} \mathbb{I}\left(Z_i \neq Z_{t,i}^{(0)}\right)$ to denote the alternative candidate $Z$-value. We estimate the $D_{t,q,i,k}^{(d)}$ by using the corresponding masked estimates $D_{t,q,i,k}^{(d,M)}$,

where

$$
\begin{aligned}
D_{t,q,i,k}^{(0,M)} &= \mathbb{E}_{(\hat{\boldsymbol{\theta}}_{t,q})}\left[\Delta_{i,k} \,|\, \mathcal{D}_{t,i}, i \in \mathcal{M}_t\right] = \mathbb{P}\Big(\Delta_{i,k} = 1 \,|\, \widetilde{Z}_{t,i}, \boldsymbol{x}_i, \hat{\boldsymbol{\theta}}_{t,q}\Big) \\
&= \frac{(f_k(Z_i \,|\, \boldsymbol{x}_i, \hat{\boldsymbol{\theta}}_{t,q}) + f_k(\check{Z}_i \,|\, \boldsymbol{x}_i, \hat{\boldsymbol{\theta}}_{t,q}))\pi_k(\boldsymbol{x}_i, \hat{\boldsymbol{\theta}}_{t,q})}{\sum_{k'=1}^{K}(f_{k'}(Z_i \,|\, \boldsymbol{x}_i, \hat{\boldsymbol{\theta}}_{t,q}) + f_{k'}(\check{Z}_i \,|\, \boldsymbol{x}_i, \hat{\boldsymbol{\theta}}_{t,q}))\pi_{k'}(\boldsymbol{x}_i, \hat{\boldsymbol{\theta}}_{t,q})} \\
D_{t,q,i,k}^{(1,M)} &= \mathbb{E}_{(\hat{\boldsymbol{\theta}}_{t,q})}\left[\Delta_{i,k} Z_i \,|\, \mathcal{D}_{t,i}, i \in \mathcal{M}_t\right] \\
&= \mathbb{P}\Big(\Delta_{i,k} = 1 \,|\, \widetilde{Z}_{t,i}, \boldsymbol{x}_i, \hat{\boldsymbol{\theta}}_{t,q}\Big) \times \mathbb{E}_{(\hat{\boldsymbol{\theta}}_{t,q})}\Big[\Delta_{i,k} \times Z_i \,|\, \widetilde{Z}_{t,i}, \Delta_{i,k} = 1, \boldsymbol{x}_i, \hat{\boldsymbol{\theta}}_{t,q}\Big] \\
&\quad + \mathbb{P}\Big(\Delta_{i,k} = 0 \,|\, \widetilde{Z}_{t,i}, \boldsymbol{x}_i, \hat{\boldsymbol{\theta}}_{t,q}\Big) \times \mathbb{E}_{(\hat{\boldsymbol{\theta}}_{t,q})}\Big[\Delta_{i,k} \times Z_i \,|\, \widetilde{Z}_{t,i}, \Delta_{i,k} = 0, \boldsymbol{x}_i, \hat{\boldsymbol{\theta}}_{t,q}\Big] \\
&= D_{t,q,i,k}^{(0,M)} \times \frac{Z_i f_k(Z_i \,|\, \boldsymbol{x}_i, \hat{\boldsymbol{\theta}}_{t,q}) + \check{Z}_i f_k(\check{Z}_i \,|\, \boldsymbol{x}_i, \hat{\boldsymbol{\theta}}_{t,q})}{f_k(Z_i \,|\, \boldsymbol{x}_i, \hat{\boldsymbol{\theta}}_{t,q}) + f_k(\check{Z}_i \,|\, \boldsymbol{x}_i, \hat{\boldsymbol{\theta}}_{t,q})} \\
D_{t,q,i,k}^{(2,M)} &= \mathbb{E}_{(\hat{\boldsymbol{\theta}}_{t,q})}\left[\Delta_{i,k} Z_i^2 \,|\, \mathcal{D}_{t,i}, i \in \mathcal{M}_t\right] \\
&= \mathbb{P}\Big(\Delta_{i,k} = 1 \,|\, \widetilde{Z}_{t,i}, \boldsymbol{x}_i, \hat{\boldsymbol{\theta}}_{t,q}\Big) \times \mathbb{E}_{(\hat{\boldsymbol{\theta}}_{t,q})}\Big[\Delta_{i,k} \times Z_i^2 \,|\, \widetilde{Z}_{t,i}, \Delta_{i,k} = 1, \boldsymbol{x}_i, \hat{\boldsymbol{\theta}}_{t,q}\Big] \\
&\quad + \mathbb{P}\Big(\Delta_{i,k} = 0 \,|\, \widetilde{Z}_{t,i}, \boldsymbol{x}_i, \hat{\boldsymbol{\theta}}_{t,q}\Big) \times \mathbb{E}_{(\hat{\boldsymbol{\theta}}_{t,q})}\Big[\Delta_{i,k} \times Z_i^2 \,|\, \widetilde{Z}_{t,i}, \Delta_{i,k} = 0, \boldsymbol{x}_i, \hat{\boldsymbol{\theta}}_{t,q}\Big] \\
&= D_{t,q,i,k}^{(0,M)} \times \frac{Z_i^2 f_k(Z_i \,|\, \boldsymbol{x}_i, \hat{\boldsymbol{\theta}}_{t,q}) + \check{Z}_i^2 f_k(\check{Z}_i \,|\, \boldsymbol{x}_i, \hat{\boldsymbol{\theta}}_{t,q})}{f_k(Z_i \,|\, \boldsymbol{x}_i, \hat{\boldsymbol{\theta}}_{t,q}) + f_k(\check{Z}_i \,|\, \boldsymbol{x}_i, \hat{\boldsymbol{\theta}}_{t,q})}
\end{aligned}
$$

### A.3.2. M-step

In the M-step, we aim to estimate a maximising $\hat{\boldsymbol{\theta}}_{t,q+1} := \arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta} \,|\, \hat{\boldsymbol{\theta}}_{t,q})$ conditional on the $E$-step estimates. First, we establish a decomposition of $Q(\boldsymbol{\theta} \,|\, \hat{\boldsymbol{\theta}}_{t,q})$. Expanding $\log L_c(\boldsymbol{\theta})$:

$$
\begin{aligned}
\log L_c(\boldsymbol{\theta}) &= \log\left\{\prod_{i=1}^{n}\left(\mathbb{P}(Z_i \,|\, \Gamma_i = k, \boldsymbol{\theta})\,\mathbb{P}(\Gamma_i = k \,|\, \boldsymbol{\theta})\right)^{\mathbb{I}(\Gamma_i = k)}\right\} \\
&= \log\left\{\prod_{i=1}^{n}\left[\prod_{k=1}^{K}\left\{\pi_{i,k}\phi\left(Z_i \,|\, \beta_{k,0} + \beta_k^T \boldsymbol{x}_i, \sigma_k^2\right)\right\}^{\mathbb{I}(\Gamma_i = k)}\right]\right\} \quad \text{[See Eq (1.2), (1.6)]}
\end{aligned}
$$

Applying the definition of $\Delta_{i,k}$ and the expanding further:

$$\begin{aligned}\log L_c(\boldsymbol{\theta}) &= \sum_{i=1}^{n} \log \left[ \prod_{k=1}^{K} \left\{ \pi_{i,k}^{\Delta_{i,k}} \phi \left( Z_i \,|\, \beta_{k,0} + \beta_k^T \boldsymbol{x}_i, \sigma_k^2 \right)^{\Delta_{i,k}} \right\} \right] \\ &= \sum_{i=1}^{n} \sum_{k=1}^{K} \Delta_{i,k} \log \left( \pi_{i,k} \right) - \frac{1}{2} \sum_{i=1}^{n} \sum_{k=1}^{K} \Delta_{i,k} \log 2\pi\sigma_k^2 \\ &\quad - \frac{1}{2} \sum_{i=1}^{n} \sum_{k=1}^{K} \frac{1}{\sigma_k^2} \left( \Delta_{i,k} (Z_i^2 + (\boldsymbol{x}_i^T \beta_k + \beta_{k,0})^2 - 2Z_i (\boldsymbol{x}_i^T \beta_k + \beta_{k,0}) \right) \end{aligned} \tag{A.5}$$

Using this, we decompose the conditional expectation of the proposed log-likelihood, following the idea of Eq (11) in [HC19] and using the expression for $\log L_c(\boldsymbol{\theta})$ in Eq (A.5) as

$$Q(\boldsymbol{\theta} \,|\, \hat{\boldsymbol{\theta}}_{t,q}) := \mathbb{E} \left[ \log PL_c(\boldsymbol{\theta}) \,|\, \mathcal{D}_t, \hat{\boldsymbol{\theta}}_{t,q} \right] = Q(\boldsymbol{w} \,|\, \hat{\boldsymbol{\theta}}_{t,q}) + \sum_{k=1}^{K} Q_k(\boldsymbol{\theta}_k \,|\, \hat{\boldsymbol{\theta}}_{t,q})$$

with the gating network component

$$Q(\boldsymbol{w} \,|\, \hat{\boldsymbol{\theta}}_{t,q}) := \sum_{i=1}^{n} \sum_{k=1}^{K} \tau_{i,k}^{(q)} \log \pi_{i,k} - \sum_{k=1}^{K-1} \gamma_k \|\boldsymbol{w}_k\|_1,$$

and $K$ expert network components

$$\begin{aligned} Q_k(\boldsymbol{\theta}_k \,|\, \hat{\boldsymbol{\theta}}_{t,q}) := &- \frac{1}{2\sigma_k^2} \left\{ \sum_{i=1}^{n} \left( D_{i,k}^{(2)} - 2D_{i,k}^{(1)} (\boldsymbol{x}_i^T \beta_k + \beta_{k,0}) + D_{i,k}^{(0)} (\boldsymbol{x}_i^T \beta_k + \beta_{k,0})^2 \right) + \lambda_k \sigma_k^2 \|\beta_k\|_1 \right\} \\ &- \frac{\log 2\pi\sigma_k^2}{2} \sum_{i=1}^{n} D_{i,k}^{(0)} \end{aligned}$$

where $\boldsymbol{\theta}_k := (\beta_{k,0}, \boldsymbol{\beta}_k, \sigma_k^2)$ captures the parameters of the $k$th expert. We can maximise each sub-function in isolation as follows.

### Gating Network

To maximise $Q(\boldsymbol{w} \,|\, \hat{\boldsymbol{\theta}}_{t,q})$ we can substitute the $\tau_{i,k}^{[q]}$ terms in Eq (B.1—B.5), Section 3.2 of [HC19] with the estimates $D_{t,q,i,k}^{(0)}$ and apply either the proximal Newton or proximal Newton-type coordinate ascent algorithms specified in the paper, which identifies a locally maximising coefficient vector $\boldsymbol{w}_{t,q+1}^*$.

### Expert Network

To maximise the marginal function $Q_k(\boldsymbol{\theta}_k \,|\, \hat{\boldsymbol{\theta}}_{t,q})$ with respect to $\boldsymbol{\beta}_k$, we follow the methodology of Section 2.4.1—2 in [HTW15]. The procedure is to iterate over the $j$ coefficients in $\beta_k$, computing individual updates using the partial derivative of $Q_k$ while fixing the remaining parameters $\boldsymbol{\beta}_{k,-j}$.[1]

In the below calculations, I use $\boldsymbol{D}_k^{(0)} := \left(D_{t,q,i',k}^{(0)}\right)_{i'=1}^{n}$ as shorthand to denote an $n \times 1$ column vector of E-step estimates, and likewise for $\boldsymbol{D}_k^{(1)}$ and $\boldsymbol{D}_k^{(2)}$. I also use $\hat{\sigma}^2, \hat{\beta}_k, \hat{\beta}_{k,0}$ to indicate other parameter estimates used in the calculations, which I differentiate from the derived update formulas $(\sigma^*)^2, \beta_k^*, \beta_{k,0}^*$. I also remove the subscripts $t, q$ from these terms for notational simplicity.

#### *Expert Coefficient Updates*

Taking a partial derivative:

$$\frac{\partial Q_k(\boldsymbol{\theta}_k \,|\, \hat{\boldsymbol{\theta}})}{\partial \beta_k} = -\frac{1}{\hat{\sigma}_k^2}\left\{X^T(\boldsymbol{D}_k^{(0)} \circ (X\beta_k)) - X^T\boldsymbol{D}_k^{(1)} + X^T\boldsymbol{D}_k^{(0)}\hat{\beta}_{k,0} + \lambda_k\hat{\sigma}_k^2\boldsymbol{g}_k\right\}$$

where $\circ$ denotes the Hadamard product i.e. element-wise multiplication, and $\boldsymbol{g}_k$ is the set of subgradients of $\|\beta_k\|_1$. Now define the partial residuals $\boldsymbol{r}_j := \boldsymbol{D}_k^{(1)} - \boldsymbol{D}_k^{(0)} \circ (X_{-j}\beta_{k,-j}) - \boldsymbol{D}_k^{(0)}\beta_{k,0}$ (as in [HTW15]). For optimality, $\mathbf{0}$ must be a subdifferential of $Q_k(\boldsymbol{\theta}_k \,|\, \hat{\boldsymbol{\theta}})$, i.e.

$$0 \in \frac{\partial Q_k(\boldsymbol{\theta}_k \,|\, \hat{\boldsymbol{\theta}})}{\partial \beta_{k,j}} \quad \forall\, j \in \{1, \ldots, p\}$$

This is equivalent to

$$0 \in -\boldsymbol{x}_j^T\boldsymbol{r}_j + \boldsymbol{x}_j^T(\boldsymbol{D}_k^{(0)} \circ (\boldsymbol{x}_j\beta_{k,j})) + \lambda_k\hat{\sigma}_k^2 \begin{cases} -1 & \beta_{k,j} < 0 \\ [-1, 1] & \beta_{k,j} = 0 \\ 1 & \beta_{k,j} > 0 \end{cases},$$

[1] In this section, a $-j$ subscript denotes the vector or matrix obtained by removal of the $j$th feature vector.

which has the solution

$$\beta_{k,j}^* = \frac{1}{\boldsymbol{x}_j^T(\boldsymbol{D}_k^{(0)} \circ \boldsymbol{x}_j)} \begin{cases} \boldsymbol{x}_j^T \boldsymbol{r}_j + \lambda_k \hat{\sigma}_k^2 & \boldsymbol{x}_j^T \boldsymbol{r}_j < -\lambda_k \hat{\sigma}_k^2 \\ 0 & |\boldsymbol{x}_j^T \boldsymbol{r}_j| \leq \lambda_k \hat{\sigma}_k^2 \\ \boldsymbol{x}_j^T \boldsymbol{r}_j - \lambda_k \hat{\sigma}_k^2 & \boldsymbol{x}_j^T \boldsymbol{r}_j > \lambda_k \hat{\sigma}_k^2 \end{cases} .$$

The hybrid function can be further simplified using the soft-thresholding function

$$S(x \mid \lambda) := \operatorname{sgn}(x) \max(0, |x| - \lambda)$$

into the final marginal coefficient update formula

$$\beta_{k,j}^* = \frac{S(\boldsymbol{x}_j^T \boldsymbol{r}_j \mid \lambda_k \hat{\sigma}_k^2)}{\boldsymbol{x}_j^T(\boldsymbol{D}_k^{(0)} \circ \boldsymbol{x}_j)}$$

***Expert Intercept Updates***

Taking the partial derivative for the intercept and solving for optimality:

$$\begin{aligned} 0 &= \frac{\partial Q_k(\boldsymbol{\theta}_k \mid \hat{\boldsymbol{\theta}})}{\partial \beta_{k,0}} \\ &= -\frac{1}{2\hat{\sigma}_k^2} \left\{ \sum_{i=1}^n -2D_{i,k}^{(1)} + 2D_{i,k}^{(0)}(\boldsymbol{x}_i^T \hat{\beta}_k + \beta_{k,0}) \right\} \\ \Rightarrow \beta_{k,0}^* &= \frac{\sum_{i=1}^n D_{i,k}^{(1)} - (\boldsymbol{D}_k^{(0)})^T X \hat{\beta}_k}{\sum_{i=1}^n D_{i,k}^{(0)}} \end{aligned}$$

***Expert Variance Updates***

Taking the partial derivative for $\sigma_k^2$ and solving for optimality:

$$\begin{aligned} 0 &= \frac{\partial Q_k(\boldsymbol{\theta}_k \mid \hat{\boldsymbol{\theta}})}{\partial \sigma_k^2} \\ &= -\frac{1}{2\sigma_k^2} \sum_{i=1}^n D_{i,k}^{(0)} + \frac{1}{2(\sigma_k^2)^2} \left\{ \sum_{i=1}^n \left( D_{i,k}^{(2)} - 2D_{i,k}^{(1)}(\boldsymbol{x}_i^T \hat{\beta}_k + \hat{\beta}_{k,0}) + D_{i,k}^{(0)}(\boldsymbol{x}_i^T \hat{\beta}_k + \hat{\beta}_{k,0})^2 \right) \right\} \\ \Rightarrow (\sigma^*)_k^2 &= \frac{\left\{ \sum_{i=1}^n \left( D_{i,k}^{(2)} - 2D_{i,k}^{(1)}(\boldsymbol{x}_i^T \hat{\beta}_k + \hat{\beta}_{k,0}) + D_{i,k}^{(0)}(\boldsymbol{x}_i^T \hat{\beta}_k + \hat{\beta}_{k,0})^2 \right) \right\}}{\sum_{i=1}^n D_{i,k}^{(0)}} \end{aligned}$$

# Appendix B

# Additional Testing Results

In this section we provide additional numerical results. In addition to the models outlined in Chapter 3, the following models are also simulated:

- The covariate-adaptive multiple testing method (CAMT) introduced in [ZC20].
- The "FDRreg" model introduced in [Sco+15], using a standard Gaussian normal as the theoretical null.
- The univariate covariate"IHW" model from [Ign+16], which was run on the covariate sums $x_\circ$.
- The oracle procedure from [SC07] discussed in Section 2.3.
- The Benjamini-Hochberg (BH) procedure from [BH95], which only operates on the $p$-values and ignores the covariates.

**Figure B.3 (Global Null) Commentary**: All investigated ZAP-GMOE configurations respect the FDR control, which is evidence for its robustness in extreme cases with near-zero non-null instances to discover. There does not appear to be any particular relationship between $K$ and the observed FDR. Only the BH procedure and IHW appear to violate the FDR bound at 0.05.

**Figure B.4 (Runtime) Commentary**: This comparison is mostly to give the reader some indication of the computational cost of running this algorithm. It was expected that the finite-sample adaptive procedures ZAP-GMOE, AdaPT-GMM$_g$ and finite-sample ZAP would have large runtimes, given that they rely on frequently updating respective data models with an EM algorithm during model execution. The ZAP-GMOE model has a near-equal execution time for $n = 5000$ but gradually diverges by several orders of magnitude. The relatively straightforward computations involved in the BH, oracle, CAMT and IHW procedures naturally result in fairly rapid computations.

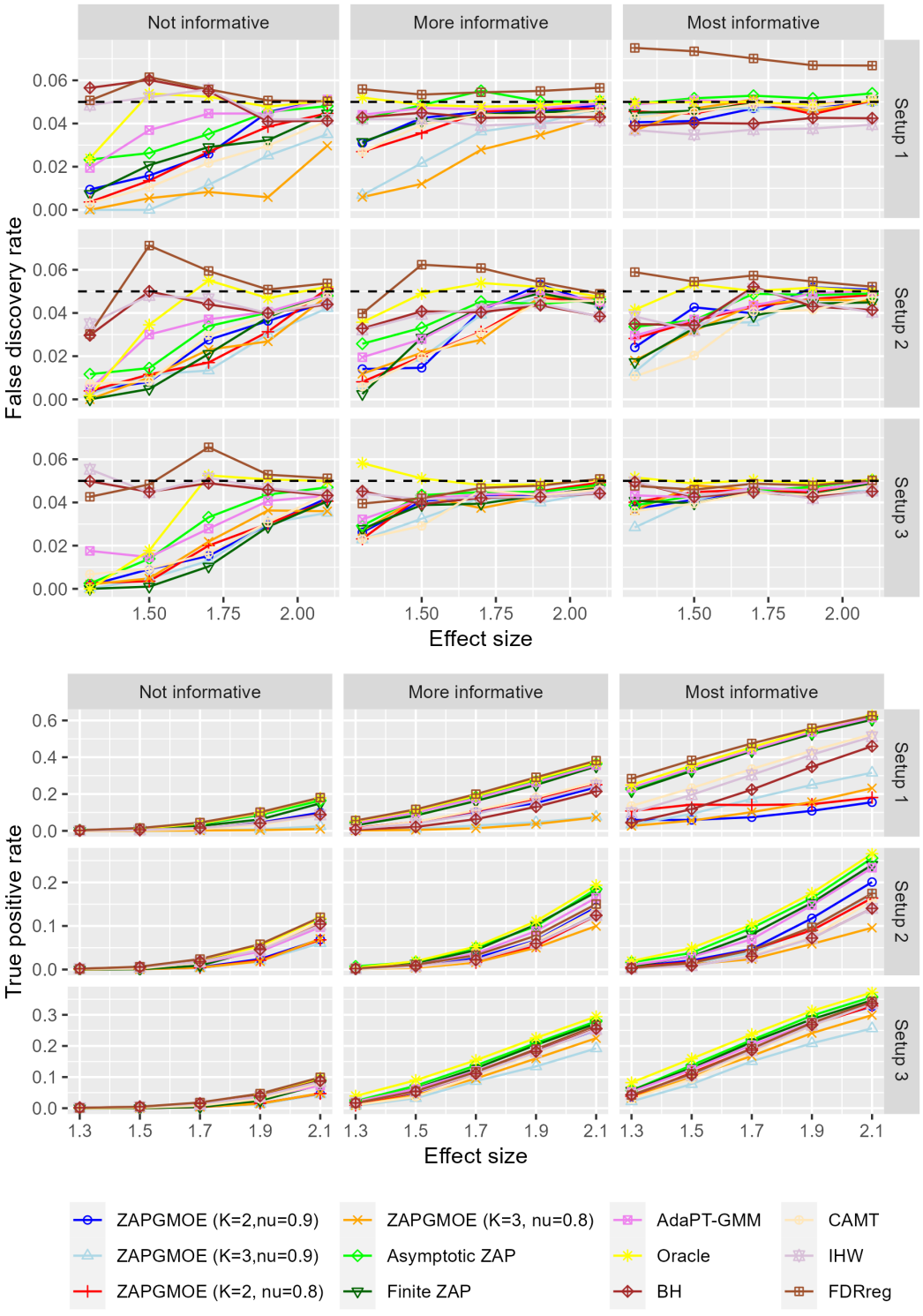



Figure B.1: (Full Baseline) The extension of Figure 3.1 to include the additional models above.

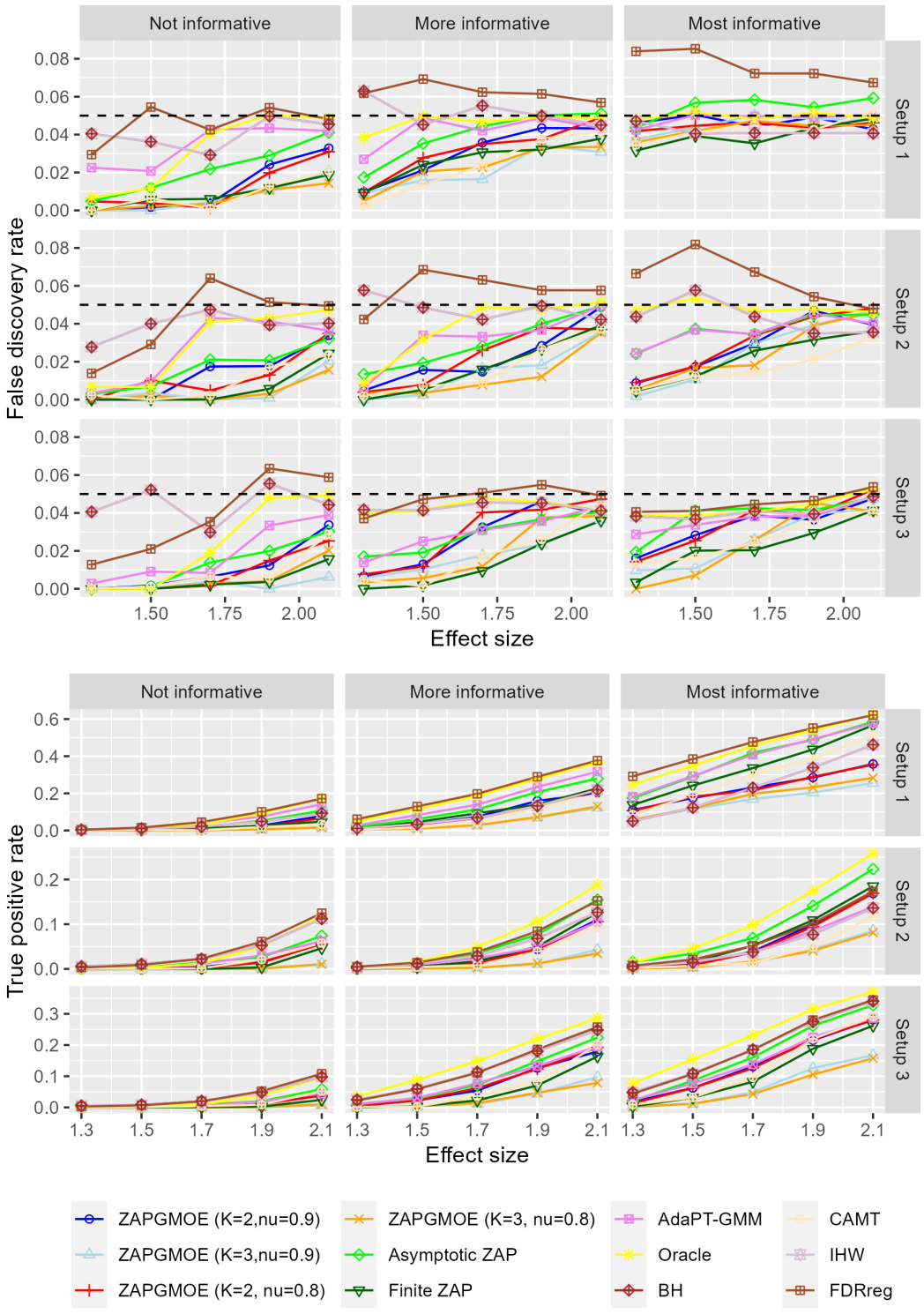



Figure B.2: (Full Smaller Baseline) The extension of Figure 3.2 to include the additional models above, evaluated on datasets with $n = 1000$.

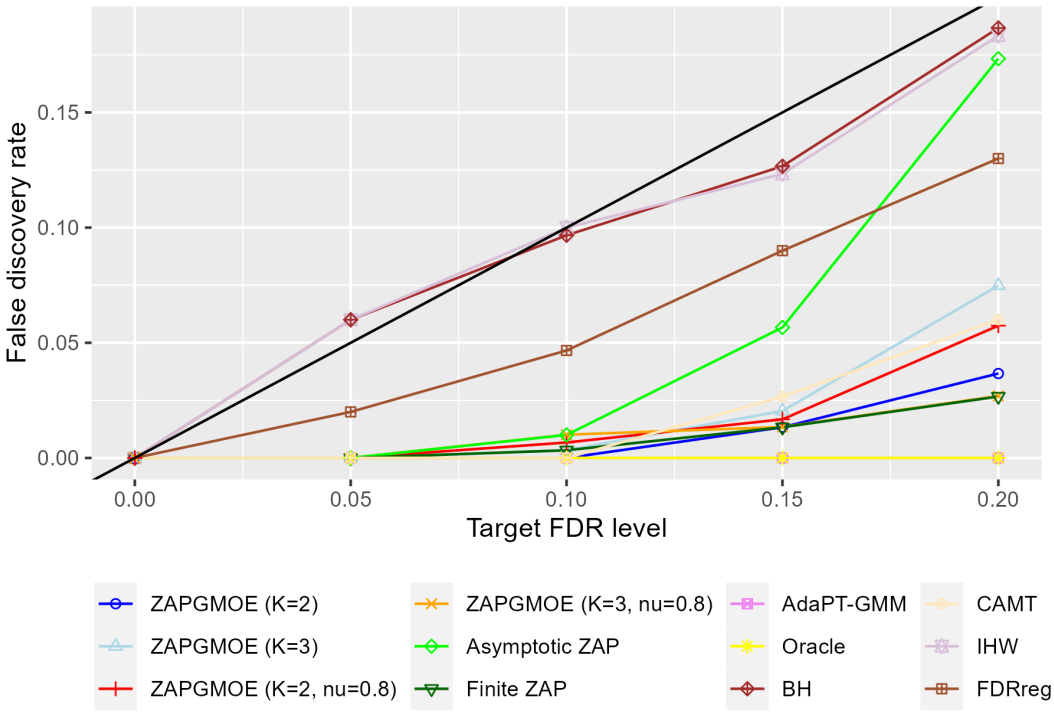



Figure B.3: (Global Null) The FDR for the model suite introduced in Chapter 3 for a fully null-distributed dataset with $n = 5000$. The target FDR level $\alpha$ is varied across $\{0, 0.05, 0.1, 0.15, 0.2\}$. The TPR was observed to be zero for all the models. The identity line in black is included for reference.

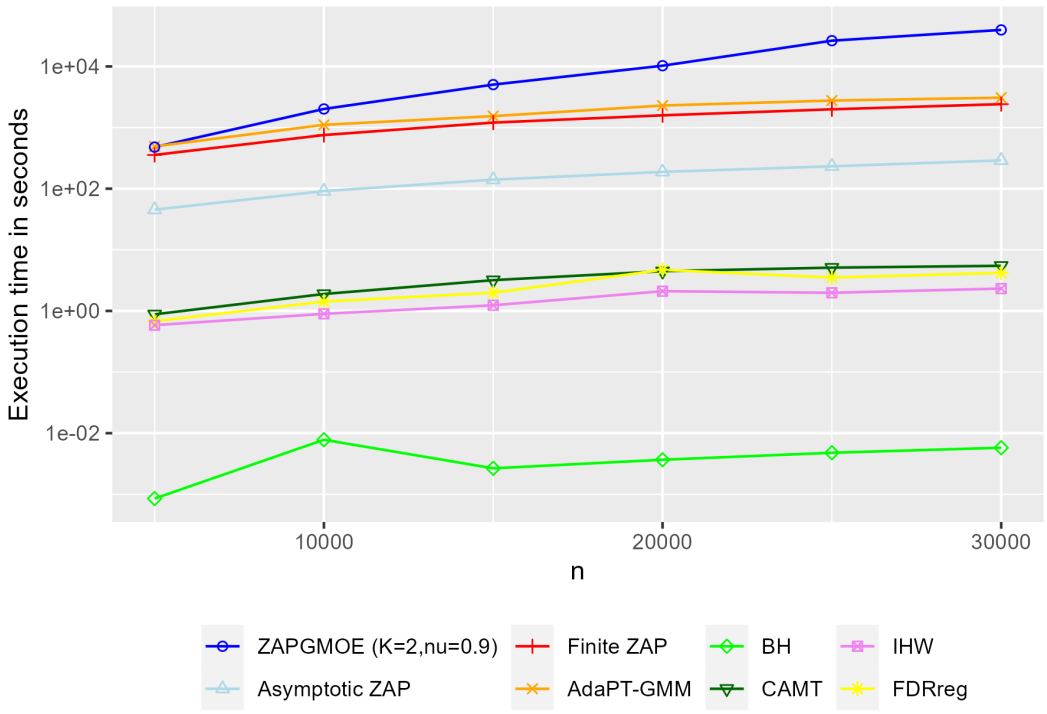



Figure B.4: (Runtime) A visualisation of the execution time in logarithmic scale for the model suite introduced in Chapter 3. We use Setup 1 with $\epsilon = 2.1, \eta = -2, \zeta = 1$ and $\sigma^2 = 1$, while varying the size $n$, and measure the execution time for a single instance.