Family-wise Error Rate Control with E-values *

Will Hartog † Lihua Lei ‡

April 8, 2025

Abstract

The closure principle is a standard tool for achieving family-wise error rate (FWER) control in multiple testing problems. In general, the computational cost for closed testing can be exponential in the number of hypotheses. The celebrated graphical approach of FWER control [Bretz et al., 2009] overcomes the computational hurdle by using weighted Bonferroni local tests on p-values with appropriately chosen weights. In this study, we extend the graphical approach to e-values. With valid e-values – common in settings of sequential hypothesis testing or universal inference for irregular parametric models – we can derive strictly more powerful local tests based on weighted averages of e-values. Consequently, this e-value-based closed test is more powerful than the corresponding graphical approach with inverse e-values as p-values. Although the computational shortcuts for the p-value-based graphical approach are not applicable, we develop efficient polynomial-time algorithms using dynamic programming for e-value-based graphical approaches with any directed acyclic graph. For special graphs, such as those used in the Holm's procedure and fallback procedure, we develop tailored algorithms with computation cost linear in the number of hypotheses, up to logarithmic factors.

1 Introduction and Preliminaries

1.1 Family-wise Error Rate

In analyses that test multiple hypotheses, it is necessary to consider the problem of multiple testing, and control for some notion of false rejection. One such approach is to control the Family-wise Error Rate (FWER), the probability of any false rejection, at a given level $\alpha \in (0, 1)$. In particular, given null hypotheses $H_1, ..., H_n$, with $\mathcal{H}_0$ denoting the subset of true nulls, an algorithm that produces the (random) rejection set $\mathcal{R}$ based on test statistics controls FWER at level α if

$$\mathbb{P}(\mathcal{R} \cap \mathcal{H}_0 \neq \emptyset) \leq \alpha.$$

*We thank Emmanuel Candès, David Siegmund, Y.J. Choe, Rianne de Heide, Lasse Fischer, Nick Koning, Ruodu Wang for helpful comments. L.L. is grateful for the support of National Science Foundation grant DMS-2338464. W.H. is supported by an Achievement Rewards for College Scientists Fellowship.

†Department of Statistics, Stanford University. Email: whartog@stanford.edu

‡Graduate School of Business and Department of Statistics, Stanford University. Email: lihualei@stanford.edu

In the literature, this property is often referred to as strong FWER control, to distinguish from to distinguish it from the weak form, which applies only when all null hypotheses are true. Since we will not address weak FWER control, except briefly in Section 2.2 to clarify existing work, we will adopt the simpler terminology throughout.

Controlling the FWER is crucial to maintain the reliability of statistical results, especially in medicine, economics, and online experimentation in tech firms. In medicine, clinical trials often compare the efficacy of several dosages or involve multiple endpoints, and FWER control ensures that no false positives undermine patient safety or lead to the adoption of ineffective therapies [Pocock et al., 1987, Bretz et al., 2009, Vickerstaff et al., 2019]. In economics, researchers often test multiple hypotheses defined by multiple interventions, multiple demographic subgroups, and multiple policy outcomes at the same time; controlling FWER helps avoid erroneous conclusions that could misguide policy decisions [Romano and Wolf, 2005, List et al., 2019, Viviano et al., 2024]. Similarly, in tech firms, A/B testing is used to compare multiple product versions or features under different metrics. Controlling FWER minimizes the chance of falsely identifying an intervention as superior, avoiding shipping ineffective features [Johari et al., 2022]. Recently, FWER control has been applied to areas beyond traditional multiple testing problems, such as distribution-free risk control for black-box machine learning algorithms [Angelopoulos et al., 2021].

1.2 Closed Testing

Given null hypotheses $H_1, ..., H_n$, a local test for index subset $I \subset [n]$, where $[n]$ is shorthand notation for the set $\{1, \ldots, n\}$, tests the hypothesis $H_I = \cap_{i \in I} H_i$ that all the nulls H_i in I are true. By convention, we call $H_1, \ldots, H_n$ elementary hypotheses.

Given a family of valid local tests $\{\phi_I\}_{I \subset [n]}$, i.e. $\mathbb{P}_{H_I}(\phi_I = 1) \leq \alpha$ for all $I \subset [n]$, the "closure" of $\{\phi_I\}_{I \subset [n]}$ rejects H_i, the elementary hypothesis, if $\phi_I = 1$ for all $I \ni i$. Any admissible procedure which controls FWER is a closed test [Sonnemann and Finner, 1988].

In the p-value literature, the most common local test is weighted Bonferroni, which given p-values $p_1, ..., p_n$, and a set of valid weights for any I, i.e. $\{w_i(I)\}$ with

$$\sum_{i \in I} w_i(I) \leq 1, \tag{1.1}$$

rejects H_I, or equivalently computes $\phi_I = 1$ iff $\min_{i \in I} \frac{p_i}{w_i(I)} \leq \alpha$, which is valid as

$$\mathbb{P}_{H_I}(\phi_I = 1) = \mathbb{P}_{H_I}(\cup_{i \in I} \{p_i \leq \alpha w_i(I)\}) \leq \sum_{i \in I} \mathbb{P}_{H_I}(p_i \leq \alpha w_i(I)) \leq \sum_{i \in I} \alpha w_i(I) \leq \alpha$$

by a union bound.

The challenge becomes simplifying the computation from the $2^n - 1$ local tests to an efficient number. The literature includes many such procedures including Holm's procedure, which uses the local test of unweighted Bonferroni with $w_i(I) = 1/|I|$.

A larger class of local tests, in which Holm's procedure belongs, is that of the graphical approach, introduced by Bretz et al. [2009]. It is a generic framework that unifies most widely-used FWER controlling procedures and allows the researcher to encode logical relationships among hypotheses. We will revisit the local tests in detail

in Section 5. Holm's procedure is an example of the graphical approach, with the complete graph and equal weights. The Fallback procedure, introduced by Wiens and Dmitrienko [2005], is an example of the graphical approach with a chain graph. In general, any graph can be used for the purposes of the graphical approach, where its closure can be efficiently computed, with a simple greedy algorithm for determining the rejection set.

1.3 E-values

E-values are a recent alternative to p-values as a tool for hypothesis testing and quantifying evidence against the null [Ramdas and Wang, 2024, Ramdas et al., 2023, Waudby-Smith and Ramdas, 2024, Wang and Ramdas, 2022]. For a null hypothesis H_0, an e-value for H_0 is a realization of an e-variable e which has the property that $\mathbb{E}_{H_0}[e] \leq 1$. By Markov's inequality, the test $\phi = 1\{e \geq 1/\alpha\}$ is valid at level α, since by Markov's inequality $\mathbb{P}(e \geq 1/\alpha) \leq \alpha$.

E-values arise naturally in sequential settings, especially in always-valid inference, where the practitioner would like the flexibility of having valid inference at any stopping time. This framework is very relevant for the online data collection in the tech setting, including the setting discussed by Johari et al. [2022]. In these settings, the e-value is often defined through an e-process $(e_{it})_{t\geq 1}$, which is a stochastic process bounded by a non-negative martingale with marginal expectation 1, known as a testing martingale [e.g. Ramdas et al., 2023].

E-values also prove to be useful in universal inference for irregular parametric models [Wasserman et al., 2020, Tse and Davison, 2022, Spector et al., 2023, Park et al., 2023]. In general, e-values offer more robustness to get valid inference in the presence of various concerns about data-dependent experimental decisions, such as a post-hoc choice of α [Grünwald, 2024, Hemerik and Koning, 2024, Koning, 2024].

1.4 Our contribution: e-value-based closed testing

In this paper, we first study closed testing procedures where each local hypothesis H_I is associated with an e-value e_I. We prove that generic e-value-based closed tests control multiple stricter forms of Type-I error than the FWER. In particular, they achieve the post-hoc FWER control introduced by Koning [2024], which allows researchers to choose the target FWER level after observing the data. In sequential settings where local e-values e_I are given by e-processes [e.g. Ramdas et al., 2023], the resulting closed test guarantees the always-valid strong FWER control: with probability $1 - \alpha$, no null hypothesis would ever be rejected if the test is applied at every time point.

Next, following the p-value-based closed testing literature, we investigate weighted e-Bonferroni local tests where the local e-value e_I is given a weighted average of e-values corresponding to elementary hypotheses. We show that the weighted e-Bonferroni tests are provably more powerful than the weighted p-Bonferroni tests with the same weights and $1/e$ p-values, the only admissible derived p-values without distributional assumptions on the e-values [Wang, 2024]. As a result, the closure of weighted e-Bonferroni tests is guaranteed to reject at least as many hypotheses as its p-value-based counterpart with $1/e$ p-values. We also examine the sequential

settings where each elementary hypothesis H_i is associated with an e-process e_{it}. In these settings, the Ville's inequality yields a more powerful p-value, known as the always-valid p-value, as the inverse of $\max_t e_{it}$, the global maximum of the e-process [Johari et al., 2022]. While $\tilde{e}_i = \max_t e_{it}$ is not a valid e-value, we show that, if the e-processes associated with elementary hypotheses are independent, the e-value-based closed test applied to pseudo e-values $\tilde{e}_i$ asymptotically controls FWER when the target level converges to zero and dominates the corresponding p-value-based closed test applied to always-valid p-values $1/\tilde{e}_i$.

Lastly, we develop efficient algorithms for e-graphical approaches, defined as the closure of weighted e-Bonferroni tests with the same weights used by p-graphical approaches and formally introduced in Section 2.4. The p-graphical approaches can be computed in polynomial time thanks to the consonance property of the weighted p-Bonferroni tests under the carefully constructed weights Gabriel [1969]. The consonance property implies a sequential rejection algorithm with $O(n^2)$ computational complexity. However, our e-value based local tests are not consonant in general, as we discuss later in Section 5. While the standard sequential rejection algorithm does not work, we develop efficient polynomial time algorithms for e-graphical approaches with arbitrary directed acyclic graphs (DAG) using dynamic programming (Section 5). The computation complexity is $O(n|\mathcal{E}|)$ in general, where $|\mathcal{E}|$ is the number of edges, but usually smaller. We also improve the algorithm for two special cases: e-Holm (Section 3) and e-Fallback (Section 4). We note that the e-Holm procedure was proposed in the prior work of Vovk and Wang [2021, 2023, 2024], though we derive derived a different representation based on the effective cutoff, enabling a direct comparison between e-Holm and p-Holm. Moreover, the previous work does not discuss other graphical approaches which pose substantially greater algorithmic challenges than e-Holm.

2 Theory of closed testing with e-values

2.1 E-value-based closed testing

In general, we define a e-value-based closed testing procedure as a collection of e-values $\{e_I\}_{I\subset[n]}$ where each e_I is a valid e-value for the intersection hypothesis $H_I = \cap_{i\in I} H_i$. The procedure rejects H_i iff $e_I \geq 1/\alpha$ for all supersets I containing i.

For p-value closed testing, it is convenient to report an adjusted p-value p_i^* for each hypothesis H_i such that H_i is rejected by the closed test iff $p_i^* \leq \alpha$. Analogously, we define the adjusted e-value

$$e_i^* \triangleq \min_{I\ni i} e_I. \tag{2.1}$$

Clearly, H_i is rejected by the e-value-based closed test iff $e_i^* \geq 1/\alpha$.

These definitions yield an immediate yet crucial result that lays the foundation of all error-controlling properties of e-value-based closed tests.

Lemma 2.1. *Let e_i^* be adjusted e-values defined in* (2.5)*. Then*

$$\max_{i\in\mathcal{H}_0} e_i^* \leq e_{\mathcal{H}_0}.$$

Since $e_{\mathcal{H}_0}$ is a valid e-value, we can easily prove the FWER controlling property.

Proposition 2.2. *The e-value-based closed test controls the FWER:*

$$\text{FWER} = \mathbb{P}\left(\max_{i\in\mathcal{H}_0} e_i^* \geq \frac{1}{\alpha}\right) \leq \mathbb{P}\left(e_{\mathcal{H}_0} \geq \frac{1}{\alpha}\right) \leq \alpha.$$

In the following two subsections, we leverage Lemma 2.1 to prove stronger error-controlling properties of e-value-based closed testing.

2.2 Post-hoc FWER Control

Grünwald [2024] formalizes the notion of roving αs, or post-hoc choice of α at which to control the error rate. They prove that if using an e-value e to test a single hypothesis, one can interpret the data at any (potentially data-dependent) level since for any $\hat{\alpha}$ which is a function of the data,

$$\mathbb{E}\left[\frac{\mathbf{1}\{e \geq \frac{1}{\hat{\alpha}}\}}{\hat{\alpha}}\right] \leq \mathbb{E}[e] \leq 1,$$

Koning [2024] generalizes the post-hoc type-I error control to multiple testing. In particular, the Appendix C.2 of Koning [2024] introduces the notion of post-hoc *weak* FWER control in terms of controlling the probability of a false rejection when all hypotheses of interest are nulls, i.e., $\mathcal{H}_0 = \{1, \ldots, n\}$.

We prove that the post-hoc *strong* FWER control can be achieved by e-value-based closed testing.

Theorem 2.3. *For closed testing adjusted e-values $\{e_i^*\}_{i=1}^n$ and set of null hypotheses $\mathcal{H}_0$, for any data-dependent function of the data $\hat{\alpha}$,*

$$\mathbb{E}\left[\frac{\mathbf{1}\{\max_{i\in\mathcal{H}_0} e_i^* \geq \frac{1}{\hat{\alpha}}\}}{\hat{\alpha}}\right] \leq 1$$

In general, a p-value defined by $\mathbb{P}[p \leq \alpha] \leq \alpha$ is not post-hoc valid, so we would expect a p-value-based closed test to not control post-hoc FWER. As a concrete example, consider a collection of independent p-values with n_0 uniform null p-values $p_i : i \in \mathcal{H}_0$. Then the p-Holm procedure rejects H_i iff its adjusted p-value $p_i^* \triangleq \max_{I\ni i} \min_{j\in I} |I| p_j$ is below α. If we choose $\hat{\alpha} = \min_i p_i^*$, the minimal level at which at least one hypothesis is rejected,

$$\begin{aligned}\mathbb{E}\left[\frac{\mathbf{1}\{\min_{i\in\mathcal{H}_0} p_i^* \leq \hat{\alpha}\}}{\hat{\alpha}}\right] = \mathbb{E}\left[\frac{1}{\min_{i\in\mathcal{H}_0} p_i^*}\right] &\geq \mathbb{E}\left[\frac{1}{n_0 \min_{j\in\mathcal{H}_0} p_j}\right] \\ &\geq \frac{1}{n_0}\frac{1}{1/(n_0+1)} = 1 + \frac{1}{n_0} > 1.\end{aligned}$$

We obtain the first inequality since by the closure principle, $\min_{i\in\mathcal{H}_0} p_j^* \leq p_{\mathcal{H}_0} = n_0 \min_{j\in\mathcal{H}_0} p_j$, and the second inequality by Jensen's inequality since the minimum of n_0 uniforms has mean $1/(n_0+1)$.

2.3 Always-valid Strong FWER Control

The closure principle also gives us an improved FWER guarantee in the always-valid setting, in which we are able to control the probability that any null hypothesis is ever falsely rejected, across time. For each hypothesis i, we define an e-process e_{it}. Then we can also define the adjusted e-processes e^*_{it} as the adjusted e-values at time t. The local test process $e_{\mathcal{H}_0,t}$ is also an e-process as an average of e-processes, so by Lemma 2.1, we have that for each t,

$$\max_{i\in\mathcal{H}_0} e^*_{it} \leq e_{\mathcal{H}_0,t}. \tag{2.2}$$

Since $e_{\mathcal{H}_0,t}$ is an e-process, by Ville's inequality, it is always-valid in the sense that

$$\mathbb{P}\left(\sup_t e_{\mathcal{H}_0,t} \geq \frac{1}{\alpha}\right) \leq \alpha.$$

This always-validity allows us to prove the following result of e-value-based closed testing.

Theorem 2.4. *For adjusted e-processes $\{e^*_{it}\}_{i\in[n],t\geq 1}$, adjusted together at each time t by e-value-based closed testing,*

$$\mathbb{P}\left(\max_{i\in\mathcal{H}_0}\sup_t e^*_{it} \geq \frac{1}{\alpha}\right) \leq \alpha$$

As a result of Proposition 2.4, the strong FWER of any null adjusted e-value *ever* being rejected is controlled by α. This means that we can make rejections based on the supremum over t over the adjusted e-values, even if the supremum was in the past.

2.4 Weighted e-Bonferroni local test

To obtain a valid local test with these e-values, one approach is to convert them into p-values using e-to-p calibrators [Vovk and Wang, 2021]. Their Proposition 2.2 shows that the unique admissible converted p-value is the inverse e-value truncated at 1, i.e., $\min\{1/e_i, 1\}$, which is further formalized in Wang [2024]. We will ignore the truncation without loss of generality, as none of the procedures discussed in this paper rejects p-values greater than or equal to 1. Then the weighted Bonferroni test rejects H_I iff

$$\min_{i\in I} \frac{1}{w_i(I)e_i} \leq \alpha \Longleftrightarrow \max_{i\in I} w_i(I)e_i \geq \frac{1}{\alpha}. \tag{2.3}$$

While (2.3) is a valid test, it does not fully unleash the potential of e-values. An appealing feature of e-values is that they combine easily; the weighted average of evalues is an e-value, as noted by Vovk and Wang [2021]. It is also possible to combine independent e-values by multiplication, which Fischer and Ramdas [2024] use with closed testing, but we do not make any independence between e-values assumptions.

This suggests an alternative local test, which we called *weighted e-Bonferroni* due to its analogousness to weighted p-Bonferroni, that rejects H_I iff

$$e_I \triangleq \sum_{i \in I} w_i(I) e_i \geq \frac{1}{\alpha}. \tag{2.4}$$

The test is valid as e_I is a valid e-value. This implies the following result.

Proposition 2.5. *If $e_1, \ldots, e_n$ are valid e-values, the e-value-based closed test, defined as the closure of weighted e-Bonferroni local tests* (2.4)*, controls FWER in finite samples.*

When the weights $w_i(I)$ are chosen according to a p-graphical approach, we call the closed test an *e-graphical approach*. The nomenclature extends to specific graphical approaches, such as e-Holm and e-Fallback.

Comparing (2.3) with (2.4), it is evident that the weighted e-Bonferroni test is more powerful in the sense that it rejects H_I whenever the weighted p-Bonferroni test (2.3) does. As a consequence, the e-value-based closed test is at least as powerful as its p-value counterpart.

The weighted e-Bonferroni local test, with weights $w_i(I)$, leads to the following optimization to compute adjusted e-values:

$$e_i^* = \min_{I \ni i} \left\{ \sum_{i \in I} w_i(I) e_i \right\}. \tag{2.5}$$

Vovk and Wang [2021, 2023] proposed an efficient algorithm for e-Holm where $w_i(I) = 1/|I|$ in our notation. We develop dynamic programming-based polynomial-time algorithms for more general e-graphical approaches.

2.5 More powerful e-value-based closed testing with e-processes

In the sequential setting discussed in Section 2.3, Johari et al. [2022] introduced the always-valid p-value, defined as $1/\tilde{e}_i$ where $\tilde{e}_i = \sup_{t \geq 0} e_{it}$. In general, $\tilde{e}_i$ is not a valid e-value – this does not contradict the admissibility of the inverse-e e-to-p calibrator [Vovk and Wang, 2021] because e-processes impose restrictions on the e-values.

When using weighted e-Bonferroni local tests, the e-value-based closed testing procedure described in Section 2.3 does not dominate closed testing based on always-valid p-values, as the latter are more powerful than inverse-e p-values. Nevertheless, we can still apply the e-value-based closed testing to the pseudo-e-values $\tilde{e}_i$. Specifically, we define

$$\tilde{e}_I = \sum_{i \in I} w_i(I) \tilde{e}_i,$$

and reject all H_i with $\tilde{e}_i^* \geq 1/\alpha$, where $\tilde{e}_i^*$ is computed according to (2.1).

We show that, when the e-processes are independent, this procedure asymptotically controls the FWER as $\alpha \to 0$.

Theorem 2.6. *For independent e-processes $\{e_{it}\}_{t \geq 0}$ across i, the closed testing adjusted pseudo-e-values asymptotically control FWER for small α:*

$$\lim_{\alpha \to 0} \frac{\mathbb{P}(\max_i \tilde{e}_i^* \geq \frac{1}{\alpha})}{\alpha} \leq 1. \tag{2.6}$$

3 e-Holm

3.1 A simple expression of the rejection rule

The Holm's procedure is the closure of a set of unweighted p-Bonferroni local tests with weight $w_i(I) = 1/|I|$ for every subset $I \subset [n]$ and $i \in I$. Following the description in Section 2.4, the e-Holm procedure closes the unweighted e-Bonferroni local tests:

$$H_I \text{ rejected iff } \frac{1}{|I|}\sum_{i\in I} e_i \geq \frac{1}{\alpha}.$$

By definition of the closure principle, the e-Holm procedure rejects H_i iff

$$\sum_{j\in I} e_j \geq \frac{|I|}{\alpha} \text{ for all } I \ni i. \tag{3.1}$$

Surprisingly, the rejection rule boils down to a simple thresholding rule with an easy-to-compute critical value, which can be computed in $O(n)$ time.

Theorem 3.1. *The e-Holm procedure rejects H_i iff*

$$e_i \geq \frac{1}{\alpha} + C, \quad \textit{where } C = \sum_{j=1}^{n} \max\left\{\frac{1}{\alpha} - e_j, 0\right\}. \tag{3.2}$$

To compute the adjusted e-values, we re-derived a $O(n \log n)$ algorithm of Vovk and Wang [2021], where the $\log n$ term is from sorting. For completeness, we restate and explain the intuition behind this algorithm in Appendix D.

3.2 Comparing e-Holm with p-Holm

The p-Holm procedure with 1/e p-values rejects $H_{(i)}$ iff

$$\frac{1}{e_{(j)}} \leq \frac{\alpha}{n-j+1}, \quad \text{for any } j \leq i.$$

The discussion in Section 2.4 already implies that (3.2) is less stringent. To bring more insight, we facilitate the comparison by considering an extreme case where the non-nulls are strong and $e_i > 1/\alpha$ almost surely (or with extremely high probability). In order for the p-Holm to make any rejection, the largest e-value $e_{(1)}$ needs to be exceeding n/α. By contrast, the threshold $1/\alpha + C$ in e-Holm (3.2) is bounded by $(|\mathcal{H}_0| + 1)/\alpha$ from above, which can be much smaller than n/α. Therefore, e-Holm can adapt to the number of nulls while p-Holm cannot. This is analogous to adaptive false discovery rate control [Storey et al., 2004, Benjamini et al., 2006].

4 e-Fallback

4.1 A naive dynamic programming algorithm

We start by presenting an suboptimal yet simple algorithm to illustrate how we apply dynamic programming to simplify the computation. The Fallback procedure

is defined by a fixed sequence of hypotheses, $H_1 \to H_2 \to \cdots \to H_n$ with associated e-values, and initial α budget $\{\alpha_i\}_{i=1}^n$ with $\sum_{i=1}^n \alpha_i = \alpha$. This is a special case of the later discussed graphical case with a chain graph. For $|I| = k$ with $I = \{i_1, ..., i_k\}$ and $i_1 < i_2 < \cdots < i_k$, the local test rejects H_I iff

$$e_I \triangleq \sum_{\ell=1}^{k} w_{i_\ell}(I) e_{i_\ell} \geq \frac{1}{\alpha}, \quad \text{where } w_{i_\ell}(I) = \sum_{i=i_{\ell-1}+1}^{i_\ell} \frac{\alpha_i}{\alpha}. \tag{4.1}$$

Above, we take $i_0 = 0$ for notational simplicity. We first prove that the adjusted e-values only depend on subsets with i being the largest element. This result extends to the general DAGs (see Theorem 5.4 in Section 5) and turns out to be a crucial property that enables the efficient dynamic programming.

Lemma 4.1. *For each $i \in [n]$,*

$$e_i^* = \max_{I \ni i, i = \max_{k \in I} k} e_I.$$

Based on this property, we give a dynamic programming algorithm with $O(n^2)$ runtime to compute the minimum over $I \ni j$ for each $j \in [n]$.

Algorithm 1: e-Fallback

Input: Vectors $\{e_i\}_{i=1}^n, \{\alpha_i\}_{i=1}^n$;
Initialize: $m_0 = 0$;
Iterate: $m_i = \min_{0 \leq j < i} \{m_j + e_i \sum_{k=j+1}^{i} \alpha_k\}$;
Output: $\{m_i\}_{i=1}^n$.

We now prove that our computation of m_i gives the adjusted e-values.

Theorem 4.2. *The output of Algorithm 1 has the property that $m_i = \alpha e_i^*$ for all $i \in [n]$.*

This dynamic programming approach performs a minimization over at most n values for each of n indices so has $O(n^2)$ runtime. However, as we will see the next two subsections, we can achieve an $O(n)$ runtime with some algorithmic shortcuts.

4.2 Speedup via a reverse Search Algorithm

In this subsection, we show that there is an algorithm to perform the optimization of e-Fallback without searching all i previous hypotheses each time we compute m_i. This algorithm will serve as a stepping stone to the graphical approaches with general DAGs in Section 5, as well as the tailored algorithm for e-Fallback in Section 4.3.

For any $i \in I \subset [i]$, instead of viewing e_I as a weighted average of e-values, we can interpret it as a weighted average of initial α-budgets $\alpha_1, \ldots, \alpha_i$, each appearing exactly once in the expression. For any such subset $I = \{i_1, \ldots, i_k\}$ with $0 = i_0 < i_1 < i_2 < \ldots < i_k = i$, it divides $[i]$ into k pieces and assigns the weight e_{i_ℓ} to the ℓ-th piece $\{\alpha_{i_{\ell-1}+1}, \ldots, \alpha_{i_\ell}\}$ in the formula of e_I. Adding an element $j \in (i_{\ell-1}, i_\ell)$ in I amounts to cut the ℓ-th chunk into two and reassigns the weight e_j to α-budgets in $\{i_{\ell-1}+1, \ldots, j\}$. To achieve the minimum, one should never add j with $e_j > e_{i_\ell}$. This observation implies a recursive structure of the optimal subset for each hypothesis.

Theorem 4.3. *For each $i \in [n]$, let $j(i) = \max\{j < i : e_j \le e_i\}$ if the set is nonempty and $j(i) = 0$ otherwise. Define I_i^* iteratively as follows:*

$$I_0^* = \emptyset, \quad I_i^* = I_{j(i)}^* \cup \{i\}.$$

Then

$$e_i^* = e_{I_i^*} = \frac{1}{\alpha}\left(\sum_{j=j(i)+1}^{i} \alpha_j\right) e_i + e_{j(i)}^*.$$

Theorem 4.3 leads to the following reverse search algorithm. Again $\{m_i/\alpha\}_{i=1}^n$ is our set of adjusted e-values.

Algorithm 2: Reverse Search for e-Fallback

Input: Vectors $\{e_i\}_{i=1}^n, \{\alpha_i\}_{i=1}^n$;
Initialize: $m_i = 0$ for all $i = 0...n$;
Iterate: For $i = 1$ to n:
Do: Set $j = i - 1$ and repeat until $e_j \le e_i$:
$m_i = m_i + \alpha_j e_i$, $j = j - 1$
$m_i = m_i + m_j$.
Output: $\{m_i\}_{i=1}^n$.

This algorithm gives a significant shortcut when the previous smaller e-value e_j is close to e_i. If, for example, we assume that the e-values are continuous and exchangeable, then the expected runtime is $O(n)$. We state this as Theorem C.1, which we prove in Section C. However, in the worst case when the e-values are decreasing, the runtime, measured by the total number of back searches, is still $O(n^2)$. In the next subsection, we will utilize a stack to recycle intermediate quantities more efficiently and reduce the computational complexity to $O(n)$.

4.3 A tailored algorithm for e-Fallback

Inspecting the construction of the optimal subset I_i^* in Theorem 4.3, we observe that iff

$$j \in I_i^* \iff e_j = \min\{e_j, \ldots, e_i\}, \tag{4.2}$$

that is, e_j is a cumulative minimum defined in a backward manner. I_i^* is the lower envelope of the first i e-values. We illustrate it in Figure 1. This implies the following iterative construction of I_i^* in a similar spirit to Lemma D.1 for e-Holm.

Lemma 4.4. *Suppose I_i^* defined in Theorem 4.3 is $\{k_1(i), \ldots, k_{m_i}(i)\}$ where $m_i = |I_i^*|$. Then $e_{k_1(i)} \le \ldots \le e_{k_{m_i}(i)}$ and*

$$I_i^* = \{i\} \cup I_{i-1}^* \setminus \{j \in I_{i-1}^* : e_j > e_i\}.$$

The construction in Lemma 4.4 can be implemented by a stack, as detailed in Algorithm 3.

As desired, Algorithm 3 has runtime $O(n)$. To see this, we break down the computation into push and pop operations. Since at most two elements are pushed

Algorithm 3: Stack Search for e-Fallback

Input: Vectors $\{e_i\}_{i=1}^n, \{\alpha_i\}_{i=1}^n$;
Initialize: $m_i = 0$ for all $i = 0...n$, stack $s = \{\}$;
Iterate: For $i = 1$ to n:
Do: Set $\alpha_i^* = \alpha_i$:
Pop (j, a_j^*) from s and add $\alpha_i^* = \alpha_i^* + \alpha_j^*$ until $e_j \leq e_i$ or $s = \{\}$
If stop because $e_j \leq e_i$: $m_i := \alpha_i^* e_i + m_j$, push (j, α_j^*) to s
If stop because $s = \{\}$: $m_i := \alpha_i^* e_i$
push (i, α_i^*) to s
Output: $\{m_i\}_{i=1}^n$.

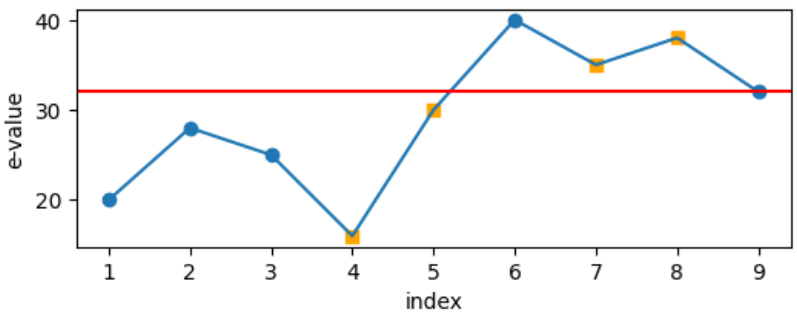

Figure 1: Visualization of stack e-Fallback Algorithm 3. For H_8, I_8^* includes all four orange dots. At step $i = 9$, the 7th and 8th dots are kicked out because their e-values are above e_9 and hence are no longer cumulative minimums. The 4th and 5th dots are kept because e_5 is the most recent dot whose e-value is below e_9.

at the end of each iteration, the total cost of push operations is $O(n)$. Using the representation (4.2), all but the last popped elements at any iteration will not be added back to the stack again. The cost of last pop adds up to $O(n)$, while the cost of all other pops combined cannot exceed the number of hypotheses n. Therefore, the overall computational complexity is $O(n)$.

5 General Approach for DAGs

5.1 Revisiting the p-graphical approach

A p-graphical approach takes as input an initial α-budget α_i for H_i, with $\sum_i \alpha_i = \alpha$ and a weighted directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with $\mathcal{V} = \{H_1, \ldots, H_n\}$, where each vertex encodes a hypothesis and the weight q_{jk} between H_j and H_k is nonnegative and satisfying $\sum_k q_{jk} \leq 1$. To define the weights $w_i(I)$ for the local test H_I, expand the graph with an additional vertex, denoted by H_{n+1}, with graph weight $q_{j(n+1)} = 1 - \sum_k q_{jk}$ and $q_{(n+1)j} = I(j = n+1)$. Consider the random walk $Z_0, Z_1, \ldots$ on the expanded graph with

$$\mathbb{P}(Z_0 = H_i) = \frac{\alpha_i}{\alpha}, \quad \mathbb{P}(Z_{t+1} = H_k \mid Z_t = H_j) = q_{jk}.$$

Then, the weight $w_i(I)$ is chosen as

$$w_i(I) = \mathbb{P}(Z_{t_I} = H_i) \tag{5.1}$$

where $t_I = \min\{t : Z_t \in \{H_i : i \in I\}\}$.

As mentioned in Section 1.4, the closure of the above local tests can be computed in polynomial time due to the consonance property, which states that, for any $I \subset [n]$, H_I is rejected if there exists $i \in I$ such that H_J is rejected for any $i \in J \subset I$. To see this, $w_i(I) \leq w_i(J)$ for any $J \subset I$. When H_I is rejected, there must be an $i \in I$ such that $p_i \leq \alpha w_i(I)$. Then, for any $i \in J \subset I$, $p_i \leq \alpha w_i(J)$ and hence H_J is rejected. The graph is allowed to be cyclical.

However, the weighted e-Bonferroni local tests are not consonant. For example, in the context of e-Holm, with three hypotheses, desired level $\alpha = 0.05$, and e-values $(e_1, e_2, e_3) = (25, 25, 10)$, the subset $\{1, 2, 3\}$ is rejected as the average e-value is $e_{\{1,2,3\}} = 20 \geq \frac{1}{\alpha}$. However, there is no i that satisfies the definition of the consonance property; in particular, $e_{\{1,3\}} = e_{\{2,3\}} = 17.5 < \frac{1}{\alpha}$.

5.2 Dynamic programming for e-DAG graphical approaches

For a DAG, the hitting probabilities can be simplified using the unidirectional flow of the graph. To facilitate the computation, for any subset I define the set of paths from $j \notin I$ to $i \in I$ through the complement of I:

$$\mathcal{P}_{j,i}^{(I)} = \{(i_0, ..., i_k) : k \geq 0, i_0 = j, i_k = i, i_\ell \notin I \ \forall \ell \in [k-1], q_{i_{\ell-1}, i_\ell} > 0 \ \forall \ell \in [k]\}. \tag{5.2}$$

Note the positivity of a graph weight ($q_{i_{\ell-1}, i_\ell} > 0$) indicates the presence of an edge. Implicit in this definition is that $i \in I, j \notin I$. Also define $k(p) := k$ the highest index of a path $p = (i_0, ..., i_k) \in \mathcal{P}_{j,i}^{(I)}$. Then,

$$w_i(I) = \frac{1}{\alpha}\left(\alpha_i + \sum_{j \notin I} \alpha_j \sum_{p \in \mathcal{P}_{j,i}^{(I)}} \prod_{\ell=1}^{k(p)} q_{i_{\ell-1}, i_\ell}\right). \tag{5.3}$$

We may think of the local test weight corresponding to redistributing the weight (α_j/α) from each excluded vertex j proportionally according to the graph structure.With the weight $w_i(I)$ defined in (5.3), the local test for e-DAG graphical approaches rejects H_I iff

$$e_I \triangleq \sum_{i \in I} w_i(I) e_i \geq \frac{1}{\alpha}. \tag{5.4}$$

For a general DAG, we notice a local phenomenon that we claim will extend to an algorithm that computes the global minimum for any index i. We first establish an equivalence between the local test e-value e_I as a weighting of e-values e_i for $i \in I$, and a weighting of alpha values α_j for $j \in [n]$.

Definition 5.1. For any subset $I \subset [n]$ and hypothesis index $j \in [n]$, define

$$e_j^{(I)} \triangleq \begin{cases} \sum_{i \in I} e_i \left(\sum_{p \in \mathcal{P}_{j,i}^{(I)}} \prod_{\ell=1}^{k(p)} q_{i_{\ell-1}, i_\ell} \right) & j \notin I \\ e_j & j \in I \end{cases} \tag{5.5}$$

When no path exists, we take $e_j^{(I)} = 0$.

Intuitively, we give this definition to give structure to the way in which an excluded vertex $i \notin I$ will have its alpha-budget α_j redistributed to the e-values in I via a reweighting based on the graph structure. Formally, we prove this relationship in Lemma 5.2.

Lemma 5.2. *For $e_j^{(I)}$ as defined in Definition 5.1 and subset $I \subset [n]$, we have*

$$e_I = \sum_{i \in I} w_i(I) e_i = \frac{1}{\alpha} \sum_{j=1}^{n} \alpha_j e_j^{(I)} \tag{5.6}$$

The result of Lemma 5.2 is useful as it shows that if we can find a subset $I \ni i$ that minimizes $e_j^{(I)}$ for all j, then we have found the minimizing subset, $\operatorname{argmin}_{I \ni i} e_I$.

To facilitate our upcoming proof of our algorithm's validity, we also show the following Lemma, which shows that we may recursively define $e_j^{(I)}$ by its children's values.

Lemma 5.3. *For $j \notin I$,*

$$e_j^{(I)} = \sum_{(j,k) \in E} q_{j,k} e_k^{(I)} \tag{5.7}$$

We assume that the nodes are already topologically ordered with respect to the input graph, which is a reasonable constraint given in practice a practitioner will be designing the graph themselves so will be able to design the data collection to have the nodes ordered properly.

We give the extension of the reverse search Fallback algorithm from Section 4.2 to the general DAG in Algorithm 4. The intuition is that we can look solely at the ancestor graph, and search backward to decide whether or not to include each node, thereby reconstructing the minimizing subset for each node i.

Algorithm 4: e-DAG Graphical Approach

Input: Vectors $\{e_i\}_{i=1}^n, \{\alpha_i\}_{i=1}^n$, transition matrix $(q_{ij}) \in \mathbb{R}^{n\times n}$, where $[n]$ is a topological ordering with respect to the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$;

Iterate: For $i \in [n]$:

Do: Initialize e-assignments $e_j^{(i)} = 0$ $(j \in [n])$, $m_i = 0$, compute A_i the set of ancestors of i, including i;

Initialize $e_i^{(i)} = e_i$, $m_i = \alpha_i e_i$:

For $i \neq j \in A_i$ decreasing $e_j^{(i)} = \min\left(e_j, \sum_{(j,k)\in E, k\in A_i} q_{jk} e_k^{(i)}\right)$ and set $m_i = m_i + \alpha_j e_j^{(i)}$.

Output: $\{m_i\}_{i=1}^n$

Theorem 5.4. *In the context of Algorithm 4, for every index i and subset $I \ni i$, $e_j^{(I)} \geq e_j^{(i)}$. Moreover, $e_j^{(i)} = e_j^{(I_i^*)}$ for some $I_i^* \ni i$, which is therefore the minimizing subset. As a result, $e_i^* = m_i/\alpha$.*

In Algorithm 4, for each node i, we recompute the e-value assignments for all $j \leq i$, each of which is a comparison of e_j to a weighting of its children's assignments. The total computational complexity is $O\left(\sum_i N_i\right)$ where N_i is the number of the edges in the ancestor graph A_i. This yields a worst-case complexity bound $O(n|\mathcal{E}|) = O(n^3)$ because $N_i \leq |\mathcal{E}| = O(n^2)$. This worst-case bound can be improved for special graphs. For example, when $\mathcal{G}$ is a tree graph, N_i is simply the depth of node H_i. For a balanced binary tree, $\max_i N_i = O(\log n)$, and thus the complexity is $O(n \log n) \ll O(n|\mathcal{E}|)$. It may be possible to further improve the complexity by re-using intermediate calculations like Algorithm 3. For example, in the tree setting, each reverse search is on a chain, so each reverse search can use the shortcut from Algorithm 2, and in particular we believe one can use a depth-first search along with a stack, following Algorithm 3, to achieve an $O(n)$ runtime. We leave further developments for future research.

6 Simulation Results

6.1 Sequential Setting

In this section we develop a framework for comparing the power of our e-value procedures to that of the corresponding p-value procedures. Without further restrictions, Vovk and Wang [2021] suggest that the only valid p-values are given by inverse e-values. Thus, our e-graphical approaches are strictly more powerful than the corresponding p-graphical approaches. To make the comparison more informative, we work in the sequential testing setting, where we can derive a valid p-value that is more powerful than the inverse e-value [Johari et al., 2022].

We present results for power in terms of how often the e-procedure improves on the p-procedures, as well as by what percentage in terms of ratio of stopping times. Both give useful information for gauging how meaningful the potential improvement of e-Holm is in the sequential setting. We examine the Holm's procedure described in Section 3 and the graphical procedure described in Section 5, specifically for a useful structuring of hypotheses in a factorial design.

For both of our simulation settings, we will be operating in a setting of a continuous stream of Gaussian data $Y_{it} \overset{i.i.d.}{\sim} \mathcal{N}(\mu_i, 1)$, with known variance 1. For arm i, i.i.d. data comes in at times $t = 1, 2, ...$, with the same mean μ_i. In Section 6.2, we will be testing each arm separately, while in Section 6.3 we will be testing differences of means of arms. In either case, the way we will construct test martingales is through the sequential probability ratio test (SPRT) with known alternative, a la Johari et al. [2022]. In particular, let $\mathcal{F}_t$ denote the σ-field generated by $\{(X_{1j}, \ldots, X_{nj}) : j \leq t\}$. For any given predictable sequence $\hat{\mu}_s \in \mathcal{F}_{s-1}$, let, we let

$$S_{it} = \prod_{s=1}^{t} \frac{\phi(Y_{is} - \hat{\mu}_i)}{\phi(Y_{is})} = \prod_{s=1}^{t} \exp(\hat{\mu}_i Y_{is} - 0.5\hat{\mu}_i^2), \tag{6.1}$$

where ϕ is the pdf (likelihood) of the Gaussian distribution. Clearly, S_{it} is a martingale with $\mathbb{E}[S_{it}] = 1$. By optional stopping theorem, for any stopping time T with respect to the filtration $\mathcal{F}_t$, $\mathbb{E}[S_{iT}] = 1$. As a result, S_{iT} is a valid e-value and, hence, $1/S_{iT}$ is a valid p-value. By Ville's inequality [Ville, 1939], $1/\max_{t \leq T} S_{it}$ is also a valid p-value. Johari et al. [2022] call it an always-valid p-value. We will perform e-graphical approaches on S_{iT} and p-graphical approaches on $1/S_{iT}$ and $1/\max_{t \leq T} S_{it}$, where T is an appropriately defined stopping time, adapted to $\mathcal{F}_t$.

The multiple testing procedures, when applied to the e-processes, provide always-valid FWER control, similar to reasoning from Johari et al. [2022]. Because $\{S_{it}\}_{t \geq 1}$ is a valid martingale adapted to $\mathcal{F}_t$ for all $i \in [n]$, the stopped processes $\{S_{iT}\}_{i \in [n]}$ are e-values and the closure principle will provide FWER control. Any stopping criterion that uses the adjusted e-values from current or previous times t is a function of the processes, and so is still adapted to the full filtration $\mathcal{F}_t$. However, if some processes are adapted to sub-filtrations, the validity may be an issue since the stopped processes may not all be e-values. Choe and Ramdas [2024] discuss examples of encountering and correcting for this issue.

6.2 e-Holm Simulations

Our setting is 20 hypotheses $H_i : \mu_i = 0$, for i.i.d. data $Y_{it} \sim \mathcal{N}(\mu_i, 1)$, where the known alternative for each is μ_{alt}, which we vary on $\{0.5, 1, 1.5, 2\}$. For our simulation, there are 5 true alternatives. We define the test martingale by (6.1) with $\hat{\mu}_i = \mu_{alt}$ and our data collection stopping time as the time at which at least one hypothesis can be rejected, after accounting for multiplicity (via each procedure). In terms of adjusted e-values $\{e^*_{it}\}_{i=1}^n$, this stopping time is $T_e = \min\{t : \max_{i \in [n]} e^*_{it} \geq 1/\alpha\}$, and we define T_{ep} and T_p analagously for the 1/e p-values and always-valid p-values.

For each value of μ_{alt}, we run 1000 Monte Carlo simulations, where we save the stopping times T_e, T_{ep}, T_p, all computed with the same sequence of simulated data. For practical purposes there is a maximum number of iterations of 2000, though no

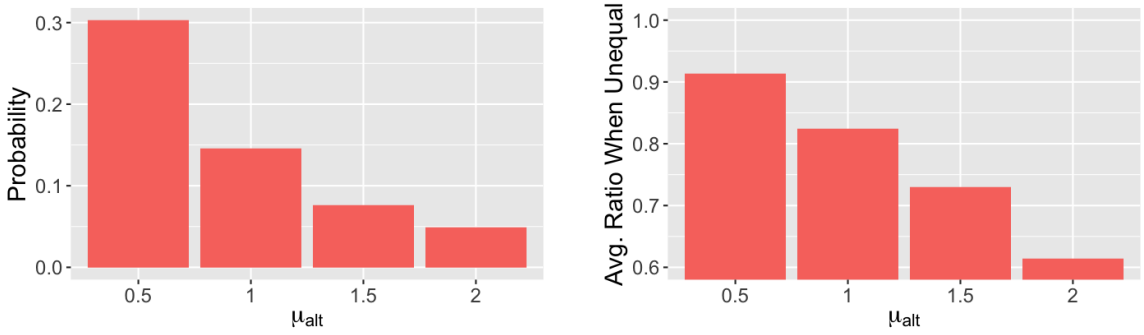

Figure 2: Left: Empirical comparison from $m = 1000$ of the probability of e-Holm improving on p-Holm. Right: Empirical ratio of stopping times, when they differ, between e-Holm and p-Holm using 1/e p-values (red) and always-valid p-values (blue). Both comparisons are for a stopping time of the first rejection out of 20 hypotheses.

run reaches this threshold. Because we stop once the most significant hypothesis is rejected, its test martingale is at its maximum, and thus $T_e \leq T_p$ always, and $T_p = T_{ep}$. This follows from the discussion of the comparison between e-Bonferroni and p-Bonferroni in Section 1.3. Thus, we only compare T_e and T_p, and compute for $m = 1000$ iterations and empirical distribution $\mathbb{P}_m$:

- Comparison (probability of improving on p-Holm): $\mathbb{P}_m(T_e < T_p)$
- Ratio (conditional on there being a difference): $\mathbb{E}_m[T_e/T_p | T_e \neq T_p]$

For the former, shown in the left panel of Figure 2, the e-Holm comparison is always non-negative, the greater the better. For the latter, shown in the right panel of Figure 2, the e-Holm ratio is always no greater than 1, the lower the better. Both metrics are necessary to understand the efficiency of e-Holm, as a small ratio is meaningless with a small comparison.

The probability comparison in Figure 2 shows that the biggest improvement in likelihood of beating p-Holm comes for smaller μ_{alt}, which makes intuitive sense as the stopping times are larger, leaving more margin for improvement. This reasoning also applies to the results in the right panel of Figure 2, which show that the lower the stopping time the lower the potential the ratio has to be, as low as 0.6 on average for $T_e \neq T_p$ when $\mu_{alt} = 2$. This intuitive rationalization aside, it is promising that all offer improvements of at least 0.05 in terms of $\mathbb{P}_m(T_e < T_p)$ and about 10% for $\mathbb{E}_m[T_e/T_p | T_e \neq T_p]$.

6.3 e-DAG Simulations

We continue our simulation study with a similar framework under a different model, this time with a structure to our hypotheses in comparison to the agnostic treatment of hypotheses in Section 6.2. We use the example of a factorial design, which is a useful framework for testing not just primary nodes, but also secondary, tertiary, etc. nodes. For example, one might be interested in the effect of a drug and the effect of gender on outcomes as primary node, but also the interaction effect of the drug for a particular gender, a secondary node.

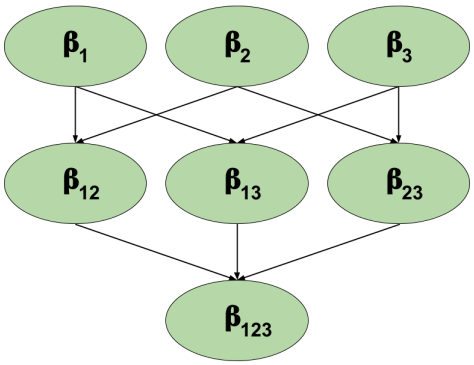

Figure 3: The transition graph for the graphical approach in a factorial design with three primary nodes.

To illustrate this framework, consider a simple three-factor model, say for A/B testing in a tech application, with three proposed treatments. The model is that data Y_i is drawn with treatments $X_i \in \{0,1\}^3$ according to

$$\begin{aligned} Y_i =& \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \beta_3 X_{i3} \\ &+ \beta_{12} X_{i1} X_{i2} + \beta_{23} X_{i2} X_{i3} + \beta_{13} X_{i1} X_{i3} + \beta_{123} X_{i1} X_{i2} X_{i3} + \epsilon_i, \end{aligned} \tag{6.2}$$

where $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$ is an i.i.d. Gaussian error term. In practice one might use a more complicated model but for simplicity we take this linear model with fixed effects and i.i.d. errors with known variance (which we take to be 1). Here the hypotheses $H_j : \beta_j = 0$ correspond to primary nodes, $H_{jk} : \beta_{jk} = 0$ correspond to secondary interaction nulls, and $H_{123} : \beta_{123} = 0$ corresponds to the tertiary interaction null.

In an experiment, the practitioner likely cares mostly about the primary hypotheses, but would like to test the secondary (and higher order) hypotheses as well. An approach is to use the graphical procedure to define a graph in which each primary hypothesis is a root, whose children are the secondary hypotheses containing the primary hypothesis. The children of H_1 are H_{12} and H_{13} in our three-way example, which is visualized in Figure 3.

In the context of the graphical approach, it is most common to take the transition probabilities to be uniform, and the starting α budget to be α/p for each of the p primary endpoints, in our case $\alpha/3$ for each of H_1, H_2, H_3. The result of this choice is that, in terms of rejecting any primary hypothesis, we use Bonferroni to determine if any node can be rejected. Once one of these nodes can be rejected, its budget transfers to the corresponding secondary hypotheses, which might be rejected, though they require a higher level of significance. The ancestor graph of H_i is just H_i, so we can only reject H_i if its standalone test martingale can reject at level $\alpha/3$.

The upside is that rejecting, for example, H_{13}, becomes potentially easier because we can use e-Bonferroni as the local test, as if H_1 is rejected, we now can boost out adjusted e-value for H_{13} via the value of H_3, if it remains unrejected. However, it is unclear a priori whether this, or the fact that the always-valid p-value gets to use the max value of the test martingale, will win out in various settings.

In order to investigate this trade-off, we perform a similar simulation study to

Section 6.2, with our three-way factorial design model. It is the same in that we have $m = 1000$ iterations, and we vary $\mu_{alt} \in \{0.5, 1, 1.5, 2\}$. Our stopping time is the rejection of H_{13}, where $\beta_1, \beta_3, \beta_{13}$ are the only nonzero parameters. We fix $\beta_1 = \beta_3 = 0.5$, with this alternative being known, and β_{13} being unknown.

We consider a process in which at time t, we receive $2^3 = 8$ measurements corresponding to draws of our model with each of $X \in \{0, 1\}^3$, which we will notate using binary. We can then isolate each coefficient (plus mean zero noise) from the data by taking combinations of our time t data. For example, $Y_{t100} - Y_{t000} \sim \mathcal{N}(\beta_1, 2)$, and similar for the other primary endpoints. Then to isolate β_{13}, we can take

$$Y_{t101} - Y_{t100} - Y_{t001} + Y_{t000} \sim \mathcal{N}(\beta_{13}, 4). \tag{6.3}$$

Because we are adding and subtracting more datapoints, the variance increases. This procedure generalizes via a pattern that follows the inclusion-exclusion principle. Then for each of these test statistics, we can construct a test martingale via a SPRT for an alternative mean, which for us is known except for H_{13}, for which we estimate $\hat{\mu}_t$ based on the sample average before time t.

With these test martingales defined, we define our stopping time to be the time that H_{13} is rejected, after accounting for multiplicity via e-DAG/ep-DAG/p-DAG, i.e. $T_e = \min\{t : e_t^{13*} \geq 1/\alpha\}$, for e_t^{13*} the adjusted e-value for H_{13}, and analagously defined for T_p and T_{ep}. Each of these procedures only depends on the test martingales for H_1, H_3, H_{13}.

As in Section 6.2, we record the probability of T_e improving on T_p, but now it could be worse, and T_p, T_{ep} could differ, so we record three comparisons: $\mathbb{P}_m(T_e < T_p), \mathbb{P}_m(T_e > T_p)$, and $\mathbb{P}_m(T_e < T_{ep})$. Since e-values have other documented pros besides our procedure (see ...), knowing the latter is relevant. We also record the ratio conditional on a difference, both for T_p and T_{ep}, because they might differ. In notation, these quantities are $\mathbb{E}_m[T_e/T_p | T_e \neq T_p]$ and $\mathbb{E}_m[T_e/T_{ep} T_e \neq T_{ep}]$.

We do two simulations. The first is for primary budget, meaning each of H_1, H_2, H_3 is assigned budget $\alpha/3$, and no budget for the secondary and tertiary hypotheses. We also perform the same analysis for equal budget among all seven hypotheses, which allows H_{13}, the subject of the simulation study, to be rejected even if H_1 and H_3 are both unrejected. This might be desirable if the secondary hypothesis is of interest separately from the primary ones. The probability comparisons are given in Figure 4 and the stopping time ratios are given in Figure 5.

In general, power tends to more favorable for e-DAG for higher values of μ_{alt}, with the only favorable ratio occurring for $\mu_{alt} = 2$. However, the results show that the gains over ep-DAG are consistent and substantial, with settings in which e-DAG is able to improve upon p-DAG.

Just like the simulations show, the equal budget experiment generally follow similar trends to plots for primary budget, but with improvements across the board. Most notably, the probability of p-DAG beating e-DAG becomes very low for $\mu_{alt} = 2$, with a corresponding ratio of less than 0.95. Both simulation settings (primary and equal budget for factorial design) show that e-DAG can consistently be a notable improvement on ep-DAG, while having mixed results compared to p-DAG, albeit with many promising settings.

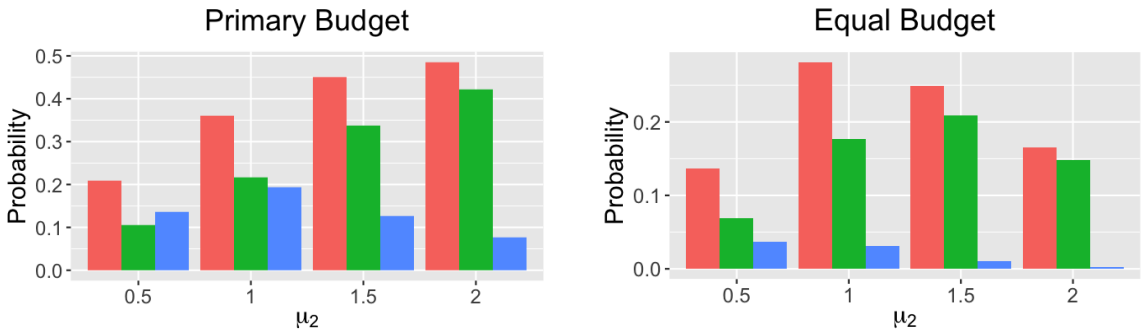

Figure 4: Empirical probability from $m = 1000$ iterations of the e-DAG stopping time being less than ep-DAG (red), less than p-DAG (green), or greater than p-DAG (blue). The stopping time is when we reject a specific secondary hypothesis in a factorial design graphical model. With primary budget and equal budget respectively.

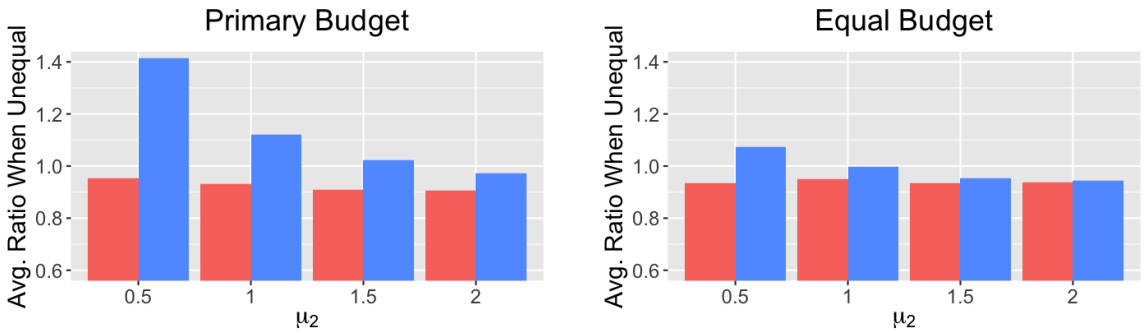

Figure 5: Empirical ratio from $m = 1000$ iterations of the stopping times, when they differ, from Figure 4 for e-DAG against p-DAG (red) and ep-DAG (blue). With primary budget and equal budget respectively.

References

Anastasios N Angelopoulos, Stephen Bates, Emmanuel J Candès, Michael I Jordan, and Lihua Lei. Learn then test: Calibrating predictive algorithms to achieve risk control. *arXiv preprint arXiv:2110.01052*, 2021.

Yoav Benjamini, Abba M Krieger, and Daniel Yekutieli. Adaptive linear step-up procedures that control the false discovery rate. *Biometrika*, 93(3):491–507, 2006.

Frank Bretz, Willi Maurer, Werner Brannath, and Martin Posch. A graphical approach to sequentially rejective multiple test procedures. *Statistics in medicine*, 28 (4):586–604, 2009.

Yuyu Chen, Peng Liu, Ken Seng Tan, and Ruodu Wang. Trade-of between validity and efficiency of merging p-values under arbitrary dependence. *Statistica Sinica*, 33(2):pp. 851–872, 2023. ISSN 10170405, 19968507. URL `https://www.jstor.org/stable/27249942`.

Yo Joong Choe and Aaditya Ramdas. Combining evidence across filtrations using adjusters. *arXiv preprint arXiv:2402.09698*, 2024.

Lasse Fischer and Aaditya Ramdas. Online closed testing with e-values. *arXiv preprint arXiv:2407.15733*, 2024.

K Ruben Gabriel. Simultaneous test procedures–some theory of multiple comparisons. *The Annals of Mathematical Statistics*, 40(1):224–250, 1969.

Peter D Grünwald. Beyond neyman–pearson: E-values enable hypothesis testing with a data-driven alpha. *Proceedings of the National Academy of Sciences*, 121 (39):e2302098121, 2024.

Jesse Hemerik and Nick W Koning. Choosing alpha post hoc: the danger of multiple standard significance thresholds. *arXiv preprint arXiv:2410.02306*, 2024.

Ramesh Johari, Pete Koomen, Leonid Pekelis, and David Walsh. Always valid inference: Continuous monitoring of a/b tests. *Operations Research*, 70(3):1806–1821, 2022.

Nick W. Koning. Post-hoc α hypothesis testing and the post-hoc p-value. *arXiv preprint arXiv:2312.08040*, 2024.

John A List, Azeem M Shaikh, and Yang Xu. Multiple hypothesis testing in experimental economics. *Experimental Economics*, 22:773–793, 2019.

Beomjo Park, Sivaraman Balakrishnan, and Larry Wasserman. Robust universal inference. *arXiv preprint arXiv:2307.04034*, 2023.

Stuart J Pocock, Nancy L Geller, and Anastasios A Tsiatis. The analysis of multiple endpoints in clinical trials. *Biometrics*, pages 487–498, 1987.

Aaditya Ramdas and Ruodu Wang. Hypothesis testing with e-values. *arXiv preprint arXiv:2410.23614*, 2024.

Aaditya Ramdas, Peter Grünwald, Vladimir Vovk, and Glenn Shafer. Game-Theoretic Statistics and Safe Anytime-Valid Inference. *Statistical Science*, 38(4):576 – 601, 2023. doi: 10.1214/23-STS894. URL `https://doi.org/10.1214/23-STS894`.

Joseph P Romano and Michael Wolf. Stepwise multiple testing as formalized data snooping. *Econometrica*, 73(4):1237–1282, 2005.

E. Sonnemann and H. Finner. Vollständigkeitssätze für multiple testprobleme. *Medizinische Informatik Und Statistik*, pages 121–135, 1988.

Asher Spector, Emmanuel Candès, and Lihua Lei. A discussion of "a note on universal inference" by tse and davison. *Stat*, 12(1):e570, 2023.

John D Storey, Jonathan E Taylor, and David Siegmund. Strong control, conservative point estimation and simultaneous conservative consistency of false discovery rates: a unified approach. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 66(1):187–205, 2004.

Timmy Tse and Anthony C Davison. A note on universal inference. *Stat*, 11(1):e501, 2022.

Victoria Vickerstaff, Rumana Z Omar, and Gareth Ambler. Methods to adjust for multiple comparisons in the analysis and sample size calculation of randomised controlled trials with multiple primary outcomes. *BMC medical research methodology*, 19:1–13, 2019.

Jean Ville. *Etude critique de la notion de collectif*. Gauthier-Villars Paris, 1939.

Davide Viviano, Kaspar Wuthrich, and Paul Niehaus. A model of multiple hypothesis testing. *arXiv preprint arXiv:2104.13367*, 2024.

Vladimir Vovk and Ruodu Wang. E-values: Calibration, combination and applications. *The Annals of Statistics*, 49(3):1736–1754, 2021.

Vladimir Vovk and Ruodu Wang. Confidence and discoveries with e-values. *Statistical Science*, 38(2):329–354, 2023.

Vladimir Vovk and Ruodu Wang. True and false discoveries with independent and sequential e-values. *Canadian Journal of Statistics*, 52(4):e11833, 2024.

Ruodu Wang. The only admissible way of merging e-values. *arXiv preprint arXiv:2409.19888*, 2024.

Ruodu Wang and Aaditya Ramdas. False discovery rate control with e-values. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 84(3):822–852, 2022.

Larry Wasserman, Aaditya Ramdas, and Sivaraman Balakrishnan. Universal inference. *Proceedings of the National Academy of Sciences*, 117(29):16880–16890, 2020.

Ian Waudby-Smith and Aaditya Ramdas. Estimating means of bounded random variables by betting. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 86(1):1–27, 2024.

Brian L. Wiens and Alexei Dmitrienko. The fallback procedure for evaluating a single family of hypotheses. *Journal of Biopharmaceutical Statistics*, 15(6):929–42, 2005.

A Proofs

A.1 Proofs of results in Section 2

Proof of Theorem 2.3. We can upper bound this quantity by replacing $\hat{\alpha}$ with α and taking the supremum, and moreover, by Lemma 2.1, we can write out

$$\mathbb{E}\left[\frac{\mathbf{1}\{\max_{i\in\mathcal{H}_0} e_i^* \geq \frac{1}{\hat{\alpha}}\}}{\hat{\alpha}}\right] \leq \mathbb{E}\left[\sup_{\alpha} \frac{\mathbf{1}\{\max_{i\in\mathcal{H}_0} e_i^* \geq \frac{1}{\alpha}\}}{\alpha}\right] \leq \mathbb{E}\left[\sup_{\alpha} \frac{\mathbf{1}\{e_{\mathcal{H}_0} \geq \frac{1}{\alpha}\}}{\alpha}\right].$$

By Grünwald [2024], this right hand quantity is bounded by 1 since $e_{\mathcal{H}_0}$ is a valid e-value and thus post-hoc valid with scaled error ≤ 1 in expectation. However, we can reconstruct a simple proof of this fact by seeing that $\mathbf{1}\{e_{\mathcal{H}_0} \geq \frac{1}{\alpha}\}/\alpha$ is maximized when the numerator is 1 and α is as large as possible, which happens when $\alpha = 1/e_{\mathcal{H}_0}$. This computes the upper bound as

$$\mathbb{E}\left[\sup_{\alpha} \frac{\mathbf{1}\{e_{\mathcal{H}_0} \geq \frac{1}{\alpha}\}}{\alpha}\right] = \mathbb{E}\left[\frac{1}{1/e_{\mathcal{H}_0}}\right] = \mathbb{E}[e_{\mathcal{H}_0}] \leq 1.$$

□

Proof of Theorem 2.4. We first swap the supremum over t and maximum over i, then appeal to the stochastic dominance (2.2).

$$\mathbb{P}\left(\max_{i\in\mathcal{H}_0} \sup_t e_{it}^* \geq \frac{1}{\alpha}\right) = \mathbb{P}\left(\sup_t \max_{i\in\mathcal{H}_0} e_{it}^* \geq \frac{1}{\alpha}\right) \leq \mathbb{P}\left(\sup_t e_{\mathcal{H}_0,t} \geq \frac{1}{\alpha}\right) \leq \alpha.$$

□

Proof of Theorem 2.6. We generate independent uniform random variables $U_i, i \in \mathcal{H}_0$. Since $\tilde{e}_i$ are independent, $\tilde{e}_{\mathcal{H}_0}$ stochastically dominates their weighted sum:

$$\tilde{e}_{\mathcal{H}_0} = \sum_{i\in\mathcal{H}_0} w_i(\mathcal{H}_0)\tilde{e}_i \succeq \sum_{i\in\mathcal{H}_0} w_i(\mathcal{H}_0)U_i^{-1}.$$

By Theorem 2 (ii) of Chen et al. [2023],

$$\lim_{\alpha\to 0} \frac{\mathbb{P}\left(\sum_{i\in\mathcal{H}_0} w_i(\mathcal{H}_0)U_i^{-1} \geq 1/\alpha\right)}{\alpha} = 1.$$

The proof is then completed by Lemma 2.1 which implies $\max_{i\in\mathcal{H}_0} \tilde{e}_i^* \leq \tilde{e}_{\mathcal{H}_0}$.

□

A.2 Proofs of results in Section 3

Proof of Theorem 3.1. We can rewrite the rejection rule (3.1) as

$$e_i \geq \frac{1}{\alpha} + \max_{I \ni i} \sum_{j \in I \setminus \{i\}} \left(\frac{1}{\alpha} - e_j \right). \tag{A.1}$$

Taking $I = \{i\}$ implies that H_i would not be rejected when $e_i < 1/\alpha$. Thus we assume $e_i \geq 1/\alpha$ throughout the rest of the proof. It is clear that the RHS of (A.1) is bounded by $1/\alpha + C$ from above. On the other hand, it equals $1/\alpha + C$ if $I = \{i\} \cup \{j : e_j < 1/\alpha\}$. The proof is then completed. □

A.3 Proofs of results in Section 4

Proof of Lemma 4.1. For any $I \ni i$, $J = I \cap [i]$ has $\alpha e_I \geq \alpha e_J$, since the α-budget for $j > i$ gets assigned nowhere. Specifically, taking $i = i_{\ell'}$ without loss of generality,

$$\alpha e_I - \alpha e_J = \sum_{\ell=\ell'+1}^{k} \left(\sum_{i=i_{\ell-1}+1}^{i_\ell} \alpha_i \right) e_{i_\ell} \geq 0.$$

□

Proof of Theorem 4.2. By Lemma 4.1, we only need to show that $m_i = \min_{i \in I \subset [i]} \alpha e_I$. For each $i = i_k$, we consider the various options for i_{k-1}, which can be any element of $[i-1]$ or $k = 1$ so $I = \{i\}$. Thus we can rewrite

$$\alpha e_i^* = \min_{i \in I \subset [i]} \alpha e_I = \min_{0 \leq j < i} \left[\min_{j \in J \subset [j]} \alpha e_J + e_i \sum_{k=j+1}^{i} \alpha_k \right],$$

where the minimum for $j = 0$ is 0 by convention. Then we use induction, with our base case of $m_1 = \alpha_1 e_1$, where we assume that $\min_{j \in J \subset [j]} \alpha e_J = m_j$ for all $j \leq i-1$, in which case we get that $\min_{i \in I \subset [i]} \alpha e_I = m_i$, as intended.

□

Proof of Theorem 4.3. We prove the result by induction. The case for $i = 1$ is evident by Lemma 4.1. Suppose the result holds for $1, \ldots, i-1$. If $j(i) = 0$, $e_i > e_j$ for all $j < i$ and $I_i^* = \{i\}$. Then, for any $i \in I \subset [i]$,

$$e_I \geq \left(\sum_{j=1}^{i} \alpha_j \right) e_i = e_{I_i^*}.$$

For the rest of the proof, we assume $j(i) > 0$.

Let $I_i = \{i_1, \ldots, i_k\}$, with $i_1 < i_2 < \ldots < i_k = i$, be any subset such that $e_i^* = e_{I_i}$. We first prove that there exists I_i' such that $e_i^* = e_{I_i'}$ and $j(i)$ is the second largest element. Assume that $j(i) \in (i_{\ell-1}, i_\ell]$ for some $\ell \leq k$, let $I_i' = \{i_1, \ldots, i_{\ell-1}, j(i), i\}$. Then

$$\alpha e_{I_i} - \alpha e_{I_i'} = \sum_{r=\ell}^{k} \left(\sum_{j=i_{r-1}+1}^{i_r} \alpha_j \right) e_{i_r} - \left(\sum_{j=i_{\ell-1}+1}^{j(i)} \alpha_j \right) e_{j(i)} - \left(\sum_{j=j(i)+1}^{i} \alpha_j \right) e_i$$

$$= \left(\sum_{j=i_{\ell-1}+1}^{j(i)} \alpha_j \right) (e_{i_\ell} - e_{j(i)}) + \left(\sum_{j=j(i)+1}^{i_\ell} \alpha_j \right) (e_{i_\ell} - e_i) + \sum_{r=\ell+1}^{k} \left(\sum_{j=i_{r-1}+1}^{i_r} \alpha_j \right) (e_{i_r} - e_i).$$

By definition of $j(i)$, $e_{i_r} > e_i \geq e_{j(i)}$ for any $r \geq \ell$. Thus, $e_{I_i} \leq e_{I_i'}$. Since I_i is a minimizer, I_i' must be so as well.

Since we can assume $j(i)$ is the second largest element in I_i, e_i^* can be written as

$$e_i^* = \frac{1}{\alpha} \left(\sum_{j=j(i)+1}^{i} \alpha_j \right) e_i + \min_{j(i) \in J \subset [j(i)]} e_J.$$

By the induction hypothesis, the second term is $e_{I^*_{j(i)}}$. Thus,

$$e_i^* = \frac{1}{\alpha} \left(\sum_{j=j(i)+1}^{i} \alpha_j \right) e_i + e^*_{j(i)}.$$

By construction, $e_i^* = e_{I_i^*}$. □

Proof of Lemma 4.4. If $j(i) = 0$, the result follows because $e_i > e_j$ for all $j < i$, $I_i^* = \{i\}$, and $I^*_{i-1} \setminus \{j \in I^*_{i-1} : e_j > e_i\} = \emptyset$. If $j(i) > 0$, we are left to show that

$$I^*_{j(i)} = \{k_1(i-1), \ldots, k_{m_i'}(i-1)\}, \quad \text{where } m_i' = \max\{m : e_{k_m(i-1)} \leq e_i\}. \tag{A.2}$$

By definition of $j(i)$, for any $j(i) < k < i$, $e_k > e_i \geq e_{j(i)}$. Thus, for each $j \leq j(i)$,

$$\min\{e_j, \ldots, e_{i-1}\} = \min\{e_j, \ldots, e_{j(i)}\}.$$

(A.2) then follows from the representation (4.2). □

A.4 Proofs of results in Section 5

Proof of Lemma 5.2. The proof is a straightforward manipulation of the expansion of e_I using the definition of $w_i(I)$, regrouping by the α_j terms.

$$\begin{aligned}
e_I &= \frac{1}{\alpha} \sum_{i \in I} \left(\alpha_i + \sum_{j \notin I} \alpha_j \sum_{p \in \mathcal{P}_{j,i}^{(I)}} \prod_{\ell=1}^{k(p)} q_{i_{\ell-1}, i_\ell} \right) e_i \\
&= \frac{1}{\alpha} \sum_{i \in I} \alpha_i e_i + \frac{1}{\alpha} \sum_{i \in I} \left(\sum_{j \notin I} \alpha_j \sum_{p \in \mathcal{P}_{j,i}^{(I)}} \prod_{\ell=1}^{k(p)} q_{i_{\ell-1}, i_\ell} \right) e_i \\
&= \frac{1}{\alpha} \sum_{i \in I} \alpha_i e_i^{(I)} + \frac{1}{\alpha} \sum_{j \notin I} \alpha_j \left(\sum_{i \in I} e_i \sum_{p \in \mathcal{P}_{j,i}^{(I)}} \prod_{\ell=1}^{k(p)} q_{i_{\ell-1}, i_\ell} \right) \\
&= \frac{1}{\alpha} \sum_{i \in I} \alpha_i e_i^{(I)} + \sum_{j \notin I} \alpha_j e_j^{(I)} = \frac{1}{\alpha} \sum_{j=1}^{n} \alpha_j e_j^{(I)}.
\end{aligned}$$

□

Proof of Lemma 5.3. To prove this, we divide the set of paths $\mathcal{P}_{j,i}^{(I)}$ into those with a given first node $i_1 = k$ which is a child of j. These will all have the same first term $q_{j,k}$. Then we know that the resulting set of paths is the same as the set $\mathcal{P}_{k,i}^{(I)}$, but with an additional index j at the start. Then we can rewrite the definition of $e_j^{(I)}$ for $j \notin I$ as

$$\begin{aligned} e_j^{(I)} &= \sum_{i \in I} e_i \left(\sum_{p \in \mathcal{P}_{j,i}^{(I)}} \prod_{\ell=1}^{k(p)} q_{i_{\ell-1}, i_\ell} \right) \\ &= \sum_{(j,k) \in E} q_{j,k} \sum_{i \in I} e_i \left(\sum_{p \in \mathcal{P}_{j,i}^{(I)}, i_1 = k} \prod_{\ell=2}^{k(p)} q_{i_{\ell-1}, i_\ell} \right) \\ &= \sum_{(j,k) \in E} q_{j,k} \sum_{i \in I} e_i \left(\sum_{p \in \mathcal{P}_{k,i}^{(I)}} \prod_{\ell=1}^{k(p)} q_{i_{\ell-1}, i_\ell} \right) = \sum_{(j,k) \in E} q_{j,k} e_k^{(I)}. \end{aligned}$$

□

Proof of Theorem 5.4. We first prove the first part by induction backward from $j = n$ to $j = 1$, for any $i \in [n]$, fixing $I \ni i$. For the base case $j = n$, we consider two cases. If $i = n$, it is clear that $e_i^{(I)} \geq e_i = e_i^{(i)}$. If $i < n$, since the hypotheses are topologically ordered, $n \notin A_i$ and hence $e_n^{(i)} = 0 \leq e_n^{(I)}$. Assuming that $e_k^{(I)} \geq e_k^{(i)}$ for all $k > j$, and we wish to show that $e_j^{(I)} \geq e_j^{(i)}$. If $j \notin A_i$ we have it automatically as $e_j^{(i)} = 0$, and otherwise,

$$e_j^{(i)} = \min \left(e_j, \sum_{(j,k) \in E} q_{jk} e_k^{(i)} \right) \leq e_j^{(I)}, \tag{A.3}$$

as

$$e_j^{(I)} = \begin{cases} e_j & j \in I \\ \sum_{(j,k) \in E} q_{jk} e_k^{(I)} & j \notin I \end{cases},$$

and $e_j^{(i)} \leq e_j$ and $e_j^{(i)} \leq \sum_{(j,k) \in E} q_{jk} e_k^{(i)} \leq \sum_{(j,k) \in E} q_{jk} e_k^{(I)}$ by the inductive assumption, meaning regardless of whether $j \in I$, $e_j^{(i)} \leq e_j^{(I)}$ as intended.

For the second part, let

$$I_i^{(i)} = \{i\}, \quad I_j^{(i)} = \begin{cases} I_{j+1}^{(i)} \cup \{j\} & e_j^{(i)} = e_j, j \in A_i \\ I_{j+1}^{(i)} & \text{otherwise} \end{cases}.$$

We then construct $I_i^* = I_1^{(i)}$. For any $j \notin I_i^*$, there is no path from j to any element in I. By Definition 5.1, $e_j^{(I_i^*)} = 0$ for any $j > i$. We prove $e_j^{(i)} = e_j^{(I_i^*)}$ for $j \in A_i$ by

induction backward from $j = i$ to $j = 1$, for each $i \in [n]$. For the base case $j = i$, $e_i^{(i)} = e_i$, which is $e_i^{(I_i^*)}$ by Definition 5.1. Assuming that $e_k^{(i)} = e_k^{(I_i^*)}$ for all $k \in A_i$ and $k > j$, we wish to prove $e_j^{(i)} = e_j^{(I_i^*)}$. If $j \in I_i^*$, $e_j^{(I_i^*)} = e_j$ by Definition 5.1 and $e_j^{(i)} = e_j$ by construction of I_i^*. If $j \notin I_i^*$,

$$e_j^{(i)} = \sum_{(j,k)\in E} q_{j,k} e_k^{(i)}.$$

By the induction hypothesis, $e_k^{(i)} = e_k^{(I_i^*)}$ for all k such that $(j, k) \in E$. By Lemma 5.3,

$$e_j^{(i)} = \sum_{(j,k)\in E} q_{j,k} e_k^{(I_i^*)} = e_j^{(I_i^*)}.$$

The proof is then completed.

□

B Extensions to Select non-DAGs

In this appendix we extend our result from Section 5 to a larger class of graphs which we call Index-Local DAGs. The key motivating observation is that our argument in Section 5 relies on the fact that for every index i, the graph over which we have to optimize is a DAG, with i as its only leaf. Thus, we define:

Definition B.1. A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is an *Index-Local DAG (ILDAG)* if for every i, the graph $\mathcal{G}^{(i)}$ formed by removing $j \in V$ such that there is no path from j to i, as well as removing any $(i, k) \in \mathcal{E}$, is a DAG.

The motivation for this is that when computing e_i^*, we have shown we only need to consider subsets of the ancestor graph $I \subset A_i$, and moreover only subsets containing i. Since $i \in I$, we know the α-budget at i will never be redistributed, i.e. $e_i^{(i)} = e_i$ in the notation of Section 5, so we can remove the edges from i and have the same minimization problem. At this point, if the remaining graph is a DAG then we have by the result of Section 5 that the DAG algorithm, for each i, will produce correct adjusted e-values $\{e_i^*\}_{i=1}^n$. This can be done by performing the DAG algorithm's backwards search for each i on its ancestor graph A_i, and ignoring any reverse edges $i \to j$.

The result is somewhat limited, but there are at least two interesting cases. The first is the cyclical Fallback graph, which is identical but for an additional edge $n \to 1$, ensuring that we are always working with the full α-budget. This graph clearly satisfies the requirement to be an ILDAG, since removing any one edge $(i \to i + 1)$ breaks the cycle and forms a Fallback chain starting at $i + 1$ and ending at i.

The other example is the gatekeeper procedure [e.g. Bretz et al., 2009] for two endpoints with a cycle between them, a visualization of which is in Figure 6.

Because we rerun Algorithm 4 for each $i \in [n]$, with potentially all nodes included in A_i, a naïve upper bound on the worst-case runtime is $O(n^2|\mathcal{E}|)$.

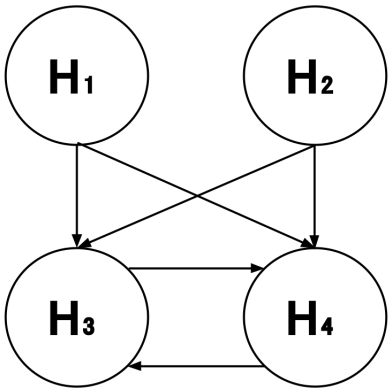

Figure 6: A gatekeeper graph with two nodes in each row. It is an example of an Index-Local DAG satisfying Definition B.1 since at least one of the two cycle-producing edges H_3, H_4 is removed when turning any given node into a leaf.

C Expected Runtime of e-Fallback Algorithm 2

Theorem C.1. *In the context of Algorithm 2:*

1. *If the e-values $\{e_i\}_{i=1}^n$ are exchangeable, then the runtime is $O_P(n)$.*
2. *If there are k non-null e-values, and the $n-k$ null e-values are exchangeable, then the runtime is $O_P(kn)$.*

Proof. First we show part (1), by denoting t_i as the number of comparisons to e_i queried, i.e. $t_i = i - j(i)$ in the notation of the previous proof. Then we see that the runtime is $O(\sum_{i=1}^n t_i)$, where in expectation we compute

$$\mathbb{E}_P t_i = \sum_{k=1}^{\infty} \mathbb{P}(t_i \geq k) = \sum_{k=1}^{i-1} \mathbb{P}(t_i \geq k), \tag{C.1}$$

where $\mathbb{P}(t_i \geq k) = \mathbb{P}(\min(\{e_{i-k+1}, ..., e_i\} = e_i) \leq \frac{1}{k!}$ with equality if the distribution of e_i is continuous, a result following from order statistics of exchangeable collections of random variables. By the Darth Vader Rule, $\mathbb{E}_P[t_i] \leq \sum_{k=1}^{i-1} \frac{1}{k!} \leq e - 1$, so we conclude that our runtime is $O_P((e-1)n) = O_P(n)$.

For part (2), we use a similar framework, but lower bound $t_i \leq n$ for e_i a non-null, and $t_i \leq k + t_i'$ where t_i' is the number of nulls preceding i until we reach a null e-value with $e_j \leq e_i$. We enumerate the nulls by indices $\{i_j\}_{j=1}^{n-k}$, and define for each $i = i_\ell$ $j_0 := \max\{j < \ell | e_{i_j} \leq e_i\}$, the corresponding index for just the nulls. Then defining $t_i' = \ell - j_0$ we have $\mathbb{E}_P[t_i'] \leq e - 1$ again, and our upper bound $t_i \leq k + e - 1$. Putting these together for k non-nulls and $n-k$ nulls, we have runtime

$$\begin{aligned} O_P(k(n) + (n-k)(k+e-1)) &= O_P(nk + nk + (e-1)n - k^2 - (e-1)k) \\ &= O_P(nk + n), \end{aligned}$$

so $O_P(nk)$ for $k > 0$.

□

This result is nice but uses exchangeability assumptions that may be unrealistic. However, it gives a sense of why we expect that this algorithm will provide a significant runtime improvement, which is achieved explicitly with the modified algorithm in Section 4.3.

D e-Holm Adjusted e-values Algorithm

While the procedure 3.1 for the rejection set has an $O(n)$ complexity, it does not compute adjusted e-values. In this appendix, we re-derive and restate the e-Holm algorithm given by Vovk and Wang [2021, 2023]. By (2.5),

$$e_i^* = \min_{I \ni i} \frac{1}{|I|} \sum_{j \in I} e_j = \min_{I \ni i} \frac{1}{|I|} \left\{ e_i + \sum_{j \in I \setminus \{i\}} e_j \right\}.$$

Clearly, the minimum is achieved by including in I the $|I| - 1$ smallest e-values other than e_i. To pin down the exact expression, we sort our e-values $e_{(1)}, ..., e_{(n)}$ in decreasing order (to match the order of the associated 1/e p-values), and call $H_{(i)}$ the hypothesis associated with the e-value $e_{(i)}$ and adjusted e-value $e^*_{(i)}$. Then

$$\min_{I \ni i, |I| = k+1} \frac{1}{|I|} \sum_{j \in I} e_{(j)} = \begin{cases} \frac{e_{(i)} + \sum_{j=n-k+1}^{n} e_{(j)}}{k+1} & k \leq n-i \\ \frac{\sum_{j=n-k}^{n} e_{(j)}}{k+1} & k > n-i \end{cases}.$$

To simplify the notation, let

$$E_k = \sum_{j=n-k+1}^{n} e_{(j)}.$$

Then we can express the adjusted e-value for $H_{(i)}$ as

$$e^*_{(i)} = \min \left\{ \min_{k \leq n-i} \frac{e_{(i)} + E_k}{k+1}, \min_{k > n-i} \frac{E_{k+1}}{k+1} \right\} = \min_{k \leq n-i} \frac{e_{(i)} + E_k}{k+1}, \tag{D.1}$$

where the second equality follows because E_k/k is non-decreasing and thus, for any $k > n - i$,

$$\min_{k > n-i} \frac{E_{k+1}}{k+1} \geq \frac{E_{n-i+1}}{n-i+1} = \frac{e_{(i)} + E_{n-i}}{n-i+1} \geq \min_{k \leq n-i} \frac{e_{(i)} + E_k}{k+1}.$$

The terms $E_1, \ldots, E_n$ can be jointly computed in $O(n \log n)$ time, including sorting and cumulative sums. Calculating all adjusted e-values separately would result in $O(n^2)$ time in total because $e^*_{(i)}$ is the minimum of $n - i$ terms. We can reduce the computation cost to $O(n)$ by leveraging the following observation.

Lemma D.1. *Let $k_i = \max\{k : e^*_{(i)} = (e_{(i)} + E_k)/(k+1)\}$. Then k_i is non-increasing in i.*

Proof. By definition of k_i, for any $k > k_i$,

$$\frac{e_{(i)} + E_k}{k+1} > \frac{e_{(i)} + E_{k_i}}{k_i+1} \Longrightarrow \left(\frac{1}{k_i+1} - \frac{1}{k+1}\right) e_{(i)} < \frac{E_k}{k+1} - \frac{E_{k_i}}{k_i+1}.$$

Since $e_{(i)} \geq e_{(i+1)}$,

$$\left(\frac{1}{k_i+1} - \frac{1}{k+1}\right) e_{(i+1)} < \frac{E_k}{k+1} - \frac{E_{k_i}}{k_i+1} \Longrightarrow \frac{e_{(i+1)} + E_k}{k+1} > \frac{e_{(i+1)} + E_{k_i}}{k_i+1}.$$

This implies the minimum of $(e_{(i+1)} + E_k)/(k+1)$ cannot be attained for some $k > k_i$. As a result, $k_{i+1} \leq k_i$. □

Lemma D.1 suggests k_i and the adjusted e-values can be computed sequentially in a decreasing order from $i = n$ to 1. The details are presented in Algorithm 5.

This algorithm correctly computes our minima as earlier argued, and has runtime $O(n \log n)$, from the sorting. The computation only changes k's value n times, and changes e^* a maximum of $2n$ times.

Algorithm 5: e-Holm Adjusted e-values

Input: Vectors $\{e_i\}_{i=1}^n$;
Sort: $\{e_i\}_{i=1}^n$ from highest to lowest as $\{e_{(i)}\}_{i=1}^n$;
Compute: $E_1 = e_{(n)}$, $E_k = E_{k-1} + e_{(n-k+1)}$ for $k = 2$ to $k = n$;
Initialize: $e^*_{(n)} = e_{(n)}$, $e^* = e_{(n)}, k = 1$;
For: $i = n - 1$ to 1,
Do: $e^* = e^* + \frac{e_{(i)} - e_{(i+1)}}{1+k}$;
While $(e^* > e_{(n-k)})$: $e^* = \frac{1+k}{2+k} e^* + \frac{1}{2+k} e_{(n-k)}$; $k = k + 1$.
$e^*_{(i)} = e^*$.
Output: $\{e^*_{(i)}\}_{j=i}^n$.