8. Supplementary Materials

8.1. *Relaxing Time-Limited Direct Effects*

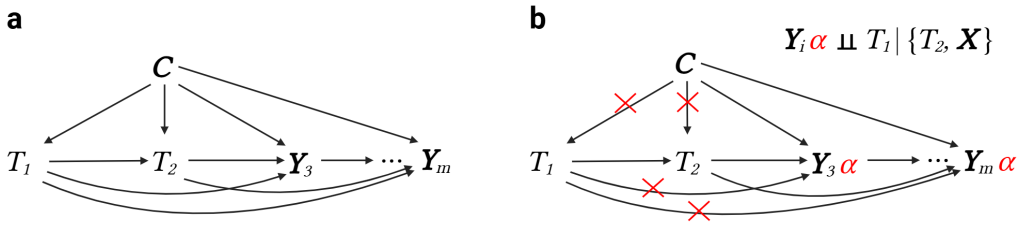

Fig. 5. **Time-limited direct effects on a subset of outcomes.** For clarity, we omit the observed confounders $\boldsymbol{X}$ from the figure. (a) T_1 has time-limited direct effects on only a subset of items in each $\boldsymbol{Y}_i$; for many other items, T_1 exerts its influence indirectly, mediated through T_2. (b) We learn non-negative weights $\alpha \geq 0$ that eliminate statistical influence from both direct causal and non-causal paths from past treatments, e.g., as indicated by the red crosses.

We can relax the assumption that time-limited direct effects apply to all items in each $\boldsymbol{Y}_i$, allowing them to instead affect only a subset of items. In Figure 5 (a), additional directed edges from T_1 to each $\boldsymbol{Y}_i$ represent scenarios where T_1 has time-limited direct effects on only a subset of the items in $\boldsymbol{Y}_i$; the effect of T_1 on $\boldsymbol{Y}_i$ for $i > 2$ operates exclusively through T_2 for the other items. Specifying the antidepressant example in the Introduction, antidepressants may not directly affect anhedonia (a specific rating scale item) a year after discontinuation in active major depression,[30] even if they can directly influence other symptoms.

Achieving conditional independence between T_1 and $\boldsymbol{Y}_i\alpha$ (after adjusting for T_2 and $\boldsymbol{X}$) accomplishes two objectives:

(a) We remove latent confounding between T_1 and each $\boldsymbol{Y}_i\alpha$, and thus between T_2 and each $\boldsymbol{Y}_i\alpha$ due to shared latent confounders with T_1.
(b) We eliminate any direct causal effect of T_1 on each $\boldsymbol{Y}_i\alpha$.

Graphically, this approach eliminates the statistical influence of both (a) non-causal paths and (b) direct causal paths from T_1 to each $\boldsymbol{Y}_i\alpha$, enabling unbiased estimation of the causal effect of T_2 on each $\boldsymbol{Y}_i\alpha$ (Figure 5 (b)).

We can formalize the relaxation by modifying the consistency and projected unconfoundedness assumptions in Section 3. Let $\boldsymbol{T} = \{T_1, \ldots, T_p\}$. We update consistency as follows:

(1) **Consistency**: If a unit receives treatments $\boldsymbol{t}$, then the observed outcome at time i coincides with the potential outcome under $\boldsymbol{t}$; that is, $\boldsymbol{Y}_i = \boldsymbol{Y}_i(\boldsymbol{t})$ when $\boldsymbol{T} = \boldsymbol{t}$.

Here, we no longer require that time-limited direct effects from treatments T_j (for all $j < p$) apply to every item in each $\boldsymbol{Y}_i$. Instead, there may exist a sparse weight vector α such that, for all t_p, the projected outcome satisfies $\boldsymbol{Y}_i(\boldsymbol{t})\alpha = \boldsymbol{Y}_i^*(t_p)$ for all treatment histories $\boldsymbol{t}$ with $T_p = t_p$. By construction, the projected outcome $\boldsymbol{Y}_i^*(t_p)$ is directly affected only by T_p; any influence of prior treatments (T_j for $j < p$) on $\boldsymbol{Y}_i^*(t_p)$ is fully mediated by T_p. As a result, earlier treatments have no direct effect on $\boldsymbol{Y}_i^*(t_p)$ once T_p is specified. We thus have the following in

the factual data for the realized treatment history $\boldsymbol{T}$:

$$\boldsymbol{Y}_i\alpha = \boldsymbol{Y}_i(\boldsymbol{T})\alpha = \boldsymbol{Y}_i^*(T_p),$$

so that the learned outcome $\boldsymbol{Y}_i\alpha$ is solely a function of T_p.

We then seek to satisfy the antecedent in the following revised projected unconfoundedness assumption:

(4) **Projected Unconfoundedness**: If there exists $\alpha \geq 0$ such that $\boldsymbol{Y}_i(\boldsymbol{T})\alpha = \boldsymbol{Y}_i^*(T_p)$ and $\boldsymbol{Y}_i^*(T_p) \perp\!\!\!\perp \{T_1, \ldots, T_{p-1}\} | T_p \cup \boldsymbol{X}$ for all $i > p$, then $\boldsymbol{Y}_i^*(t_p) \perp\!\!\!\perp T_p | \boldsymbol{X}$ for all $i > p$ and all t_p.

The first condition, $\boldsymbol{Y}_i(\boldsymbol{T})\alpha = \boldsymbol{Y}_i^*(T_p)$, ensures that the weighted outcome is only directly affected by T_p. The second condition states that the weighted outcome and historical treatments are independent after conditioning on T_p and $\boldsymbol{X}$ – ruling out any backdoor or direct causal paths between historical treatments and outcomes. As a result, the standard unconfoundedness condition is restored for $\boldsymbol{Y}_i^*(t_p)$ for all t_p, allowing unbiased estimation of the causal effect of T_p on $\boldsymbol{Y}_i\alpha$ for each time point $i > p$.

With this relaxed theoretical backbone in place, all subsequent concepts and experimental procedures described from Section 4 onward in the main text continue to apply seamlessly, with the minor adjustment that we return to the notation $\boldsymbol{Y}_i(T_p)\alpha$ to denote the empirically learned, non-negatively weighted outcome at time i under treatment T_p. This expression is equivalent to $\boldsymbol{Y}_i^*(T_p)$ as defined above in the factual data: specifically, $\boldsymbol{Y}_i^*(T_p)$ represents a projected outcome that, by construction, is directly affected only by T_p and not by earlier treatments, and is realized as a linear combination $\boldsymbol{Y}_i\alpha$ for a suitably chosen data-driven weight vector α. Consequently, all algorithms and empirical analyses involving $\boldsymbol{Y}_i\alpha$ in the main text remain valid for $\boldsymbol{Y}_i^*(T_p)$ under this generalized framework.

8.2. *Pseudocode*

Let $\mathcal{L}$ denote the objective function defined in Expression (2). We maximize $\mathcal{L}$ for a fixed λ using projected gradient ascent, as summarized in Algorithm 1, with the step size determined by backtracking line search satisfying the Armijo condition. At each iteration, the empirical gradient in Line 3 is computed in matrix notation as follows, under the assumption that all variables have already been residualized with respect to the appropriate conditioning terms:

$$\begin{aligned}\nabla_\alpha \mathcal{L} = \sum_{i=p+1}^{m} \Bigg[& \underbrace{\frac{\boldsymbol{Y}_i^T T_p}{\|\boldsymbol{Y}_i\alpha\|\|T_p\|} - \frac{(\boldsymbol{Y}_i\alpha)^T T_p}{\|\boldsymbol{Y}_i\alpha\|^3\|T_p\|}\boldsymbol{Y}_i^T\boldsymbol{Y}_i\alpha}_{(a)} \\ & \underbrace{- \frac{\lambda}{p-1}\sum_{j=1}^{p-1} 2\,\frac{(\boldsymbol{Y}_i\alpha)^T T_j}{\|\boldsymbol{Y}_i\alpha\| \cdot \|T_j\|}\left(\frac{\boldsymbol{Y}_i^T T_j}{\|\boldsymbol{Y}_i\alpha\|\|T_j\|} - \frac{(\boldsymbol{Y}_i\alpha)^T T_j}{\|\boldsymbol{Y}_i\alpha\|^3\|T_j\|}\boldsymbol{Y}_i^T\boldsymbol{Y}_i\alpha\right)}_{(b)} \\ & \underbrace{- \frac{1}{K-1}\sum_{k=1}^{K-1}\left(\frac{M_i\alpha_k}{\sqrt{\alpha_k^T M_i\alpha_k\alpha^T M_i\alpha}} - \frac{\alpha_k^T M_i\alpha}{(\alpha^T M_i\alpha)^{3/2}\sqrt{\alpha_k^T M_i\alpha_k}}M_i\alpha\right)}_{(c)} \Bigg]\end{aligned} \tag{3}$$

Algorithm 1 DEBIAS for fixed λ

Input: Treatment at the time of interest T_p; historical treatments T_j for $j < p$; outcome items $\boldsymbol{Y}_i$ for $i > p$; covariates $\boldsymbol{X}$; number of summary scores $s \leq$ cardinality of $\boldsymbol{Y}_i$

Output: α_k for all $k \leq s$

1: **for** $K \in \{1, \ldots, s\}$ **do**
2: **while** α not converged **do**
3: $\nabla f(\alpha) \leftarrow$ compute gradient for α (Equation (3))
4: $\eta \leftarrow$ backtracking line search with the Armijo condition
5: $\alpha \leftarrow \alpha + \eta \nabla f(\alpha)$
6: $\alpha \leftarrow \max(\alpha, 0)$
7: $\alpha \leftarrow \alpha / \|\alpha\|_1$
8: **end while**
9: $\alpha_K \leftarrow \alpha$
10: **end for**

The summation over Mahalanobis cosine similarities (c) is only present when $K \geq 2$. After computing the gradient, the algorithm determines an appropriate step size via backtracking line search with the Armijo condition[20] (Line 4), updates the iterate by moving in the direction of the gradient (Line 5), and then projects onto the feasible set by enforcing non-negativity (Line 6) and ℓ_1-normalization (Line 7). This procedure is repeated until convergence for each of the s summary scores.

We emphasize that Algorithm 1 is executed for a fixed value of the regularization parameter λ. The optimal value of λ is selected via cross-validation to maximize the main correlation term summed over all s scores:

$$\sum_{k=1}^{s} \sum_{i=p+1}^{m} \operatorname{cor}(\boldsymbol{Y}_i \alpha_k, T_p | T_1, \boldsymbol{X}),$$

but subject to the constraint that the geometric mean of the minimum p-values for the squared correlation coefficient (the confounding test), taken across scores and cross-validation folds, remains above a user-specified threshold ($\gamma = 0.05$ by default).

If no value of λ satisfies the confounding constraint, the algorithm can proceed in one of two ways: (1) it may abstain from producing an outcome score for the current configuration, or (2) it may select the value of λ whose associated confounding p-value is closest to, but does not exceed, the threshold γ. While abstention is preferred in practical applications to avoid unreliable inferences, we report the solution corresponding to the λ that yields a confounding p-value closest to 0.05 in the experiments for fair comparison against existing algorithms. After selecting the optimal λ via cross-validation, the final model is retrained on the full dataset using this value, and the resulting summary scores are reported.

8.3. *Computational Complexity*

We now present a detailed analysis of the computational complexity of the DEBIAS algorithm. Let n denote an upper bound on the number of subjects with complete data across all time

points, q the number of outcome items per vector $\boldsymbol{Y}_i$, r the number of covariates in $\boldsymbol{X}$, m the number of unique time points, p the index of the treatment of interest, s the number of extracted summary scores, I the maximum number of gradient ascent iterations, and L the number of line search steps per iteration. Throughout, we assume $n \gg r$.

Residualization at each time point requires $O(n(r+1)^2)$ operations, and is performed at most $(m-p)(p-1)$ times in total. Therefore, the overall complexity for residualization is $O\big((m-p)(p-1)n(r+1)^2\big)$. Calculation of correlation matrices M for all $m-p$ time points requires $O((m-p)nq^2)$ operations.

We now consider the computational cost of evaluating the gradient in Equation (3) after residualization. The main correlation term (a) requires $O(nq^2)$ time, the confounding penalty (b) requires $O((p-1)nq^2)$, and the diversity-promoting penalty (c) requires $O(sq^2)$. Each of these must be computed for each of the $m-p$ relevant time points, yielding a total cost per gradient evaluation of:

$$O\left((m-p)\left[\,p\,nq^2 + sq^2\,\right]\right).$$

Each gradient update is augmented by a backtracking line search, which entails L additional objective evaluations per iteration. Excluding preprocessing, evaluation of the objective function for a single set of weights requires $O(nq)$ for the primary correlation, $O((p-1)nq)$ for the confounding penalty, and $O(sq^2)$ for the diversity-promoting penalty. The total per-evaluation cost is thus:

$$O\left((m-p)\left[\,(p-1)\,nq + sq^2\,\right]\right).$$

and the overall per-iteration complexity is increased by a factor of L.

Consequently, for a fixed λ, performing I gradient ascent iterations for each of s summary scores results in a total complexity of:

$$O\left((m-p)(p-1)n(r+1)^2 + (m-p)nq^2 + sIL(m-p)\left[\,(p-1)\,nq + sq^2\right] + sI(m-p)\left[\,p\,nq^2 + sq^2\,\right]\right).$$

This can be equivalently expressed as:

$$O\left((m-p)\left[(p-1)\,n\,(r+1)^2 + n\,q^2\,(1+sIp) + sIL\,(p-1)\,n\,q + s^2I(L+1)q^2\right]\right).$$

In summary, the computational complexity of DEBIAS scales linearly with the number of subjects (n) and time points (m), but quadratically with the number of outcome items (q), covariates (r), and summary scores (s). In most practical settings, m and q remain moderate, and s is typically much smaller than q, so the algorithm remains computationally tractable for datasets with tens of thousands of subjects and hundreds of outcome variables. When λ is selected via cross-validation, the total computational cost increases proportionally with the number of candidate λ values times the number of cross-validation folds.

8.4. *Additional TADS Results*

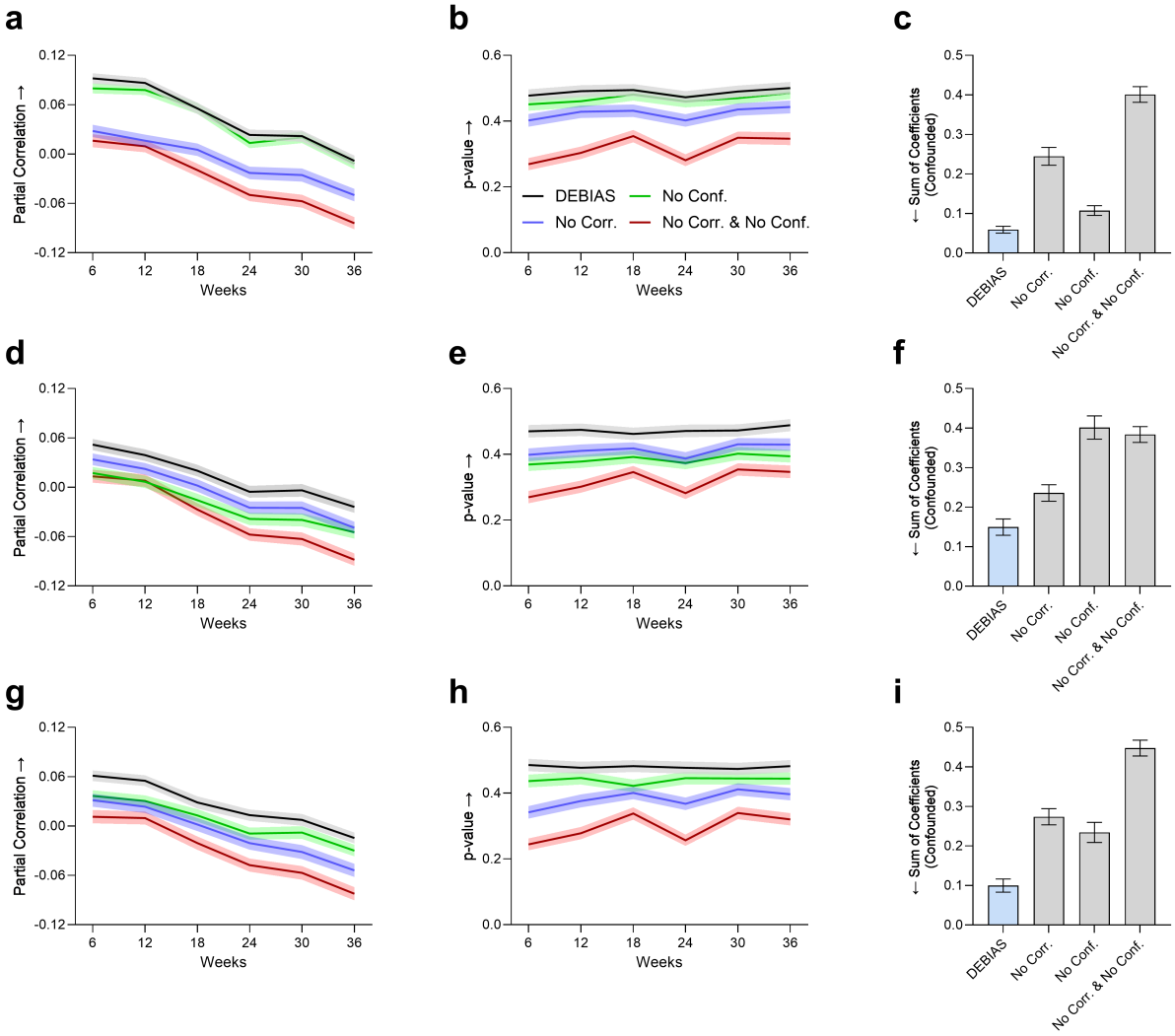

Fig. 6. **Ablation accuracy results for TADS.** We evaluated DEBIAS against three ablated variants: replacing the correlation objective with MSE (no corr.), removing the confounding penalty by setting $\lambda = 0$ (no conf.), and both replacing the correlation with MSE and removing the confounding penalty (no corr. and no conf.). The first two columns of subfigures share the same legend in panel (b). For the first recovered score, panels (a) - (c) show that DEBIAS achieved the highest correlation with the learned outcomes (a), the highest p-value for the confounding tests (b), and the lowest total weight assigned to confounded outcome items (c). DEBIAS similarly outperformed all ablated variants for the second (d-f) and third (g-i) recovered scores across all metrics. As expected, the variant with both modifications (no corr. and no conf.) consistently performed the worst across all measures. These results indicate that both correlation maximization and confounding minimization are essential for DEBIAS to achieve optimal performance.

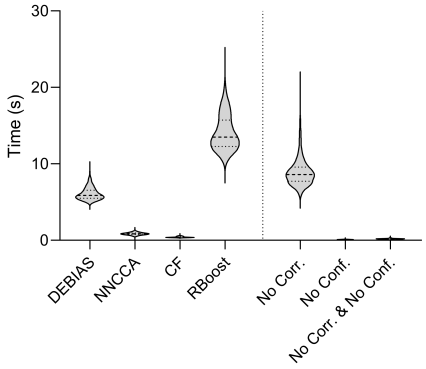

Fig. 7. **Timing results for TADS.** Violin plots showing that DEBIAS usually completed within 10 seconds, slightly faster than RBoost. Dotted lines within each violin denote the 25th, 50th and 75th percentiles.

8.5. *Additional CATIE Results*

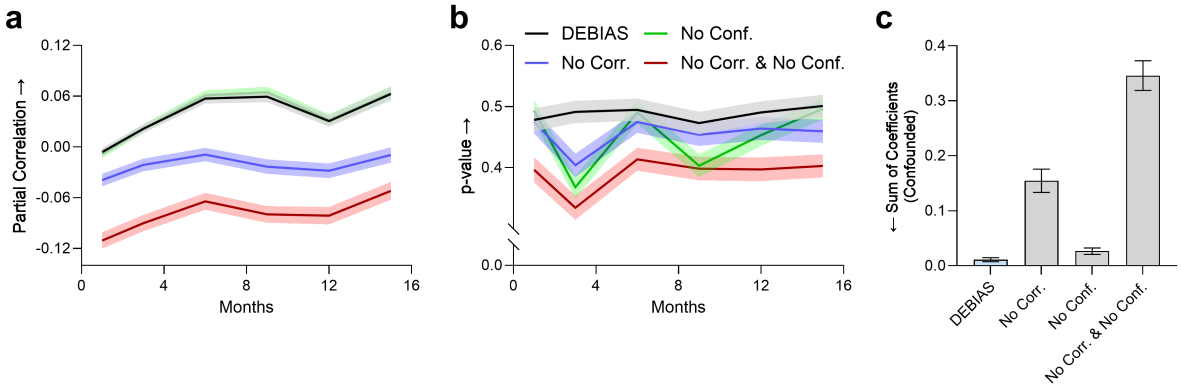

Fig. 8. **Ablation accuracy results for CATIE.** For the first recovered score, both DEBIAS and the variant without the confounding penalty (no conf.) achieved the highest correlation with treatment assignment in (a). However, DEBIAS demonstrated superior control of confounding, as indicated by the largest p-values for the squared correlation coefficient in (b) and the lowest sum of coefficients assigned to confounded outcome items in (c). For the second and third scores, all methods produced similarly low, near-zero correlations; these results are therefore not shown.

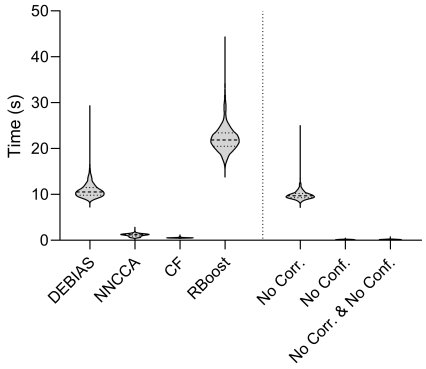

Fig. 9. **Timing results for CATIE.** DEBIAS completed within approximately 20 seconds on this dataset, outperforming RBoost in computational efficiency but requiring more time than the remaining comparator algorithms.