RETHINKING OPTIMIZATION METHODS FOR HIERARCHICAL FEATURE SELECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Rare-event forecasting often depends on interpretable conjunctions: a single keyword, location, or temporal cue may be weak, while their joint occurrence can identify a meaningful precursor. Hierarchical-Task Numerical conjunctive feature Learning (HTNL) selects Numerical Conjunctive Features (NCFs) through a structured sparsity objective. However, the resulting optimization problem is challenging, making optimization design critical to its practical performance. We give a unified diagnosis of four HTNL optimization families: original majorization-minimization (MM), variational quadratic form (VQF), alternating direction method of multipliers (ADMM), and weighted group lasso (WGL). Our analysis distinguishes three key factors that jointly determine HTNL optimization behavior. First, VQF and MM are algebraically equivalent under matched variational updates and optimize a squared-penalty formulation. Second, ADMM optimizes the unsquared objective through consensus splitting; we replace its truncated FISTA proximal solver with a scalar bisection update (ADMM-Bisect) for more accurate treatment of near-threshold blocks. Third, WGL targets the same unsquared objective, while all methods face rapidly decaying active-set admission scores. Across two synthetic datasets and 11 civil-unrest and flu-surveillance tasks, VQF achieves the highest mean AUC with the most compact non-collapsing supports. ADMM-Bisect reduces measured runtime, while calibrated WGL avoids the empty-support failures of its original scale. Synthetic results further show how proximal accuracy, pruning, and search budget affect candidate recovery, especially for longer conjunctions. Code will be made publicly available upon publication.

1 INTRODUCTION

Forecasting rare localized phenomena such as civil unrest (Compton et al., 2014a; Ramakrishnan et al., 2014; Cadena et al., 2015; Goode et al., 2015), disease outbreaks (Achrekar et al., 2011; Lamb et al., 2013), and disaster responses (Sakaki et al., 2010) requires models that can identify weak but meaningful precursor patterns in noisy spatiotemporal streams (Zhao et al., 2015; 2016; Ning et al., 2018). In these domains, a single keyword, location, or temporal cue is rarely decisive. Predictive signal often appears only through conjunctions (Zhao et al., 2019): a protest-related phrase in a particular country during a narrow time window, or a flu-related phrase co-occurring with a regional surveillance pattern. Numerical Conjunctive Features (NCFs), defined as products of primitive co-occurrence signals, give a direct and interpretable representation of such multi-factor precursors for spatial event forecasting (Zhao et al., 2019).

Hierarchical-Task Numerical conjunctive feature Learning (HTNL) (Zhao et al., 2019) addresses this problem by selecting predictive conjunctions from an exponentially large NCF space without explicitly enumerating all candidates. Its hierarchical sparsity structure provides two key benefits: it favors compact, interpretable sets of precursor conjunctions and simultaneously identifies the geographic groups in which those conjunctions are informative. This makes HTNL particularly suitable for rare-event forecasting. The difficulty is optimization. HTNL combines an outer hierarchy over NCFs with nested geographic group norms, producing strongly coupled structured-sparsity penalties for which standard lasso or group-lasso updates (Tibshirani, 1996; Yuan & Lin, 2006; Jacob et al., 2009; Jenatton et al., 2011; Bach et al., 2012) cannot be applied directly. Moreover, because the exponentially large feature space must be explored through active-set expansion, performance depends not only on solving the current optimization problem accurately but also on whether useful unseen conjunctions can enter the candidate set.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Common optimization strategies for structured sparsity include majorization–minimization, variational reformulation, operator-splitting methods such as ADMM, and weighted group-lasso continuation (Hunter & Lange, 2004; Boyd et al., 2011; Eckstein & Bertsekas, 1992; Parikh & Boyd, 2014; Friedman et al., 2010). For HTNL, we therefore compare four corresponding solvers: MM (introduced in (Zhao et al., 2019)), VQF, ADMM, and WGL continuation. MM and VQF use quadratic surrogate updates, ADMM separates the smooth loss from the nonsmooth hierarchical penalty through consensus splitting, and WGL solves a sequence of weighted group-lasso problems along a regularization path.

We diagnose these solvers at three levels: the objective formulation, the inner nonsmooth update, and the outer active-set search. At the formulation level, MM and VQF are not distinct optimization methods: under matched auxiliary variables, they yield the same weighted-ridge updates, following the broader connection between variational reformulation and majorization–minimization (Hunter & Lange, 2004). At the solver level, ADMM's truncated FISTA proximal step (Beck & Teboulle, 2009) can leave near-threshold blocks numerically nonzero. We replace this approximation with a scalar bisection solver (ADMM-Bisect), which computes the proximal update to numerical precision. WGL exhibits a different issue: its continuation path is highly sensitive to regularization scale and can collapse to an empty support when the initial penalty is too large. Beyond these solver-specific effects, all methods must discover unseen NCFs from an exponentially large candidate space, connecting HTNL to feature screening methods for large structured feature spaces (Tibshirani et al., 2012).

We evaluate these mechanisms on two synthetic datasets and 11 real-world rare-event tasks spanning Latin American civil unrest (Compton et al., 2014a; Ramakrishnan et al., 2014) and US flu surveillance (Achrekar et al., 2011; Lamb et al., 2013). The synthetic experiments isolate proximal accuracy, pruning, and search-budget effects, while the real-world tasks evaluate prediction, sparsity, and runtime. Non-collapsing HTNL variants outperform a class-weighted logistic baseline, with VQF achieving the highest mean AUC and the most compact supports. ADMM-Bisect removes proximal residuals with lower measured runtime, while regularization calibration prevents WGL from collapsing to empty supports.

This paper makes three main contributions:

1. **Unified optimization diagnosis.** We study four HTNL optimization families-MM, VQF, ADMM, and WGL continuation-on rare-event NCF selection tasks, and show how they split into equivalent, sparse-selector, over-selector, and collapse regimes.
2. **Exact ADMM proximal update.** We introduce an exact bisection proximal operator for the ADMM subproblem and combine it with KKT-pruning, yielding a more reliable and faster ADMM variant while preserving the original HTNL objective.
3. **Structural active-set bound.** We prove that the active-set KKT expansion score decays rapidly with NCF length, giving a structural explanation for the observed performance ceiling and showing that breaking this ceiling requires redesigning the active-set expansion rule.

2 Background

2.1 HTNL Model

Let $\mathbf{X} \in \mathbb{R}^{T\times S\times P}$ be the input tensor, where T is the number of time steps, S is the number of geographic locations, and P is the number of primitive features. We write $X_{s,t} \in \mathbb{R}^P$ for the feature vector at location s and time t, and $Y_{s,t}$ for the corresponding binary event label.

A Numerical Conjunctive Feature (NCF) is a product of primitive features. For an NCF $v = \{j_1, \ldots, j_k\} \subseteq [P]$, its value is

$$\phi_v(X_{s,t}) = \prod_{j\in v} x_j(s,t). \tag{2.1}$$

Let $\mathcal{V}$ be the candidate NCF set and let $D(v)$ and $A(v)$ denote the descendant and ancestor sets of v in the NCF lattice.

Hierarchical-Task Numerical conjunctive feature Learning (HTNL) learns a location-specific weight $W_{s,v}$ for each NCF v. The prediction score is

$$f_{s,t}(W) = \sum_{v\in\mathcal{V}} W_{s,v}\phi_v(X_{s,t}) + b_s. \tag{2.2}$$

The empirical loss is

$$\mathcal{L}(W) = \sum_{s=1}^{S} \sum_{t=1}^{T} C_{s,t} \ell(Y_{s,t}, f_{s,t}(W)), \tag{2.3}$$

where ℓ is a convex classification loss and $C_{s,t}$ is a sample weight.

2.2 Bi-Level Regularizer

HTNL uses a bi-level sparsity penalty to couple NCF selection and geographic structure. Let $\mathcal{G}$ be the set of geographic groups. For each NCF u, define

$$r_u(W) = \sum_{g \in \mathcal{G}} \|\{W_{s,u}\}_{s \in g}\|_2 . \tag{2.4}$$

For each ancestor v, the local hierarchical penalty is

$$\Omega_v(W) = d_v \left(\sum_{u \in D(v)} r_u(W)^p \right)^{1/p}, \qquad 1 < p \leq 2, \tag{2.5}$$

where $d_v = a^{|v|}$ is a depth-dependent penalty weight. The full HTNL regularizer is

$$\Omega(W) = \sum_{v \in \mathcal{V}} d_v \left(\sum_{u \in D(v)} \left(\sum_{g \in \mathcal{G}} \|\{W_{s,u}\}_{s \in g}\|_2 \right)^p \right)^{1/p} . \tag{2.6}$$

The learning objective is

$$\min_W \mathcal{L}(W) + \lambda \Omega(W). \tag{2.7}$$

2.3 Active-Set Framework

Because the NCF lattice is combinatorial, HTNL optimizes over a working set $A^{(m)} \subseteq \mathcal{V}$ at outer iteration m:

$$\min_{W_{A^{(m)}}} \mathcal{L}(W_{A^{(m)}}) + \lambda \Omega(W_{A^{(m)}}). \tag{2.8}$$

After solving the restricted problem, inactive candidates are scored by their KKT violation:

$$\text{score}(v) = \frac{1}{d_v} \left\| \sum_{s=1}^{S} \sum_{t=1}^{T} \nabla_W \mathcal{L}(W^*) \phi_v(X_{s,t}) \right\|_2 . \tag{2.9}$$

Candidates with $\text{score}(v) > \tau_{\text{KKT}}$ are added:

$$A^{(m+1)} = A^{(m)} \cup \{v \notin A^{(m)} : \text{score}(v) > \tau_{\text{KKT}}\}. \tag{2.10}$$

The loop stops when no new candidate violates the threshold or when the maximum number of outer rounds is reached. All solvers studied in this paper share this active-set framework and differ only in how they solve the restricted inner problem.

3 Methodology

Figure 1 summarizes the optimization pipeline. Starting from the HTNL objective in (2.7), we solve the restricted problem in (2.8) using three approaches: VQF converts the two-level regularizer in (2.6) into weighted ridge updates, ADMM retains the nonsmooth hierarchy through consensus splitting, and WGL reduces it to a weighted group-lasso path. All three feed into the same outer update in (2.10).

3.1 Variational Quadratic Form

We first derive a variational quadratic form for the HTNL regularizer. The goal is to transform the nested nonsmooth penalty in (2.6) into a weighted quadratic penalty after introducing auxiliary variables. This gives a majorize-minimize-style update: for fixed auxiliary variables, the inner optimization becomes a weighted ridge problem in W. This is the same weighted-ridge reduction used by the original MM solver in (Zhao et al., 2019), and Proposition 3.2 shows that the two formulations produce identical W-updates under the same auxiliary-weight update rule.

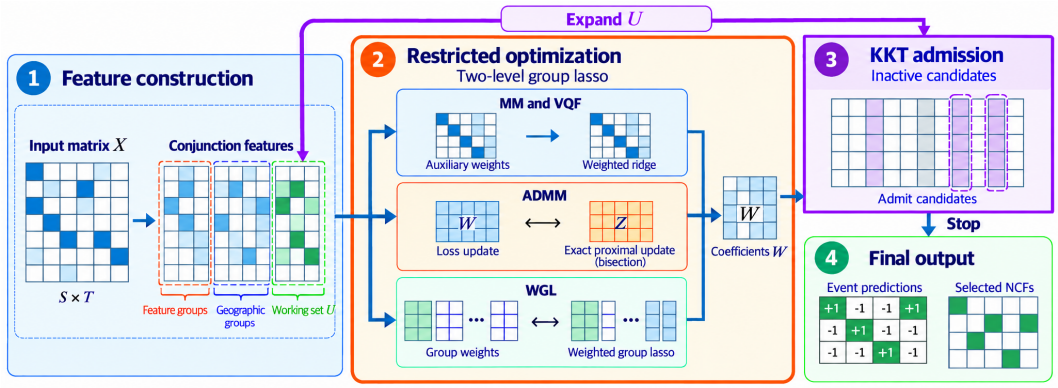


Figure 1: **Main pipeline of HTNL optimization and NCF selection.** HTNL constructs conjunction features from X and optimizes the restricted two-level group-lasso problem using MM–VQF, ADMM, or WGL. The resulting coefficients score unseen NCFs by the KKT criterion, admitted candidates expand the working set U, and the process repeats until convergence, producing event predictions and selected NCFs.

Lemma 3.1 (Variational quadratic form of a two-level group lasso)**.** *Let $W = \{W_{j,u}\}$, where $u \in \mathcal{V}$ indexes conjunctive features and $j \in g$ indexes locations in geographical group $g \in \{1, \dots, G\}$. For each $v \in \mathcal{V}$, let $D(v) \subseteq \mathcal{V}$ be the descendant set of v, and let $d_v > 0$. Recall $\Omega(W)$ has the form Equation* (2.6) *with $1 < p \leq 2$. Let*

$$\Theta_{m,r} = \left\{ x \in \mathbb{R}_+^m : \sum_{i=1}^m x_i^r \leq 1 \right\}. \tag{3.1}$$

Then $\Omega(W)^2$ admits the equivalent quadratic representation

$$\Omega(W)^2 = \min_{\gamma,\lambda,\mu} \sum_{v\in\mathcal{V}} \frac{d_v^2}{\gamma_v} \sum_{u\in D(v)} \frac{1}{\lambda_{v,u}} \sum_{g=1}^{G} \frac{1}{\mu_{u,v,g}} \sum_{j\in g} \|W_{j,u}\|_2^2, \tag{3.2}$$

where $\gamma \in \Theta_{|\mathcal{V}|,1}, \lambda_v \in \Theta_{|D(v)|,\bar{p}}, \mu_{u,v} \in \Theta_{G,1}$, $\bar{p} = \frac{p}{2-p}$. Thus, for fixed γ, λ, μ, the regularizer is a weighted ridge penalty in W.

The proof is deferred to Appendix C.1.

Proposition 3.2 (Equivalence of VQF and MM)**.** *Let $\Omega(W)$ be the HTNL regularizer in* (2.6)*, and let its variational quadratic representation be given by* (3.2)*. Suppose VQF and MM start from the same initialization $W^{(0)}$ and update the auxiliary variables (γ, λ, μ) by the same exact minimization of the variational representation at each iterate. Then, for every iteration $t \geq 0$, the two methods solve the same weighted-ridge subproblem and produce identical iterates:*

$$W_{\mathrm{VQF}}^{(t)} = W_{\mathrm{MM}}^{(t)}. \tag{3.3}$$

The proof is deferred to Appendix C.2; algebraically, the two methods yield identical auxiliary weights and therefore the same weighted-ridge update at every iteration.

3.2 Direct ADMM Optimization

We next consider ADMM, which directly optimizes the nonsmooth two-level group-lasso objective rather than replacing it with a quadratic surrogate. On a fixed candidate set U, ADMM introduces consensus copies $Z_{v,u,g} = P_{u,g} W_U$ to separate the smooth loss from the hierarchical penalty (Eckstein & Bertsekas, 1992; Boyd et al., 2011; Parikh & Boyd, 2014). The W-update is therefore smooth, while the original nested regularizer is retained in the Z-update.

Lemma 3.3 (Consensus decomposition of the HTNL objective)**.** *Let $D_U(v) = D(v) \cap U$ and define the local two-level penalty $\Omega_v(Z_v) = \left(\sum_{u\in D_U(v)} \left(\sum_{g\in\mathcal{G}} \|Z_{v,u,g}\|_2 \right)^p \right)^{1/p}$. Introducing $Z_{v,u,g} = P_{u,g} W_U$ gives the ADMM updates*

$$W_U^{k+1} = \arg\min_{W_U} \mathcal{L}(W_U) + \frac{\rho}{2} \sum_{v\in U} \sum_{u\in D_U(v)} \sum_{g\in\mathcal{G}} \|P_{u,g} W_U - Z_{v,u,g}^k + Y_{v,u,g}^k\|_2^2. \tag{3.4}$$

$$Z_v^{k+1} = \mathrm{prox}_{(\lambda d_v/\rho)\Omega_v}(Q_v^{k+1}), \qquad Q_{v,u,g}^{k+1} = P_{u,g} W_U^{k+1} + Y_{v,u,g}^k. \tag{3.5}$$

Thus, ADMM isolates the nonsmooth hierarchy in a local proximal problem while leaving the parameter update smooth.

The complete consensus formulation, dual update, and stopping conditions are given in Appendix B.1.

The update in (3.5) is exact only when its proximal problem is solved exactly. In practice, however, the published ADMM implementation (Zhao et al., 2019) approximates this update using a fixed number of FISTA iterations (Beck & Teboulle, 2009). We next quantify the resulting finite-step error and show why it can affect NCF support recovery near the zero/nonzero boundary.

3.3 Finite-Step Proximal Error in ADMM

Although (3.5) defines the exact proximal update, the published implementation approximates it with a fixed number of FISTA iterations (Beck & Teboulle, 2009). To quantify the resulting error, consider the consensus form

$$\mathcal{L}_\rho(W, Z, \Lambda) = \mathcal{L}(W) + \lambda\Omega^2(Z) + \frac{\rho}{2}\|W - Z + \Lambda\|_2^2. \tag{3.6}$$

For fixed W and Λ, the exact Z-update is

$$Z^* = \mathrm{prox}_{\lambda\Omega^2/\rho}(Q) = \arg\min_Z \left\{ \lambda\Omega^2(Z) + \frac{\rho}{2}\|Z - Q\|_2^2 \right\}, \qquad Q = W + \Lambda. \tag{3.7}$$

Let $Z^{(K)}$ be the output after K FISTA iterations. Define the proximal objective and its finite-step error as

$$\mathcal{F}_\rho(Z) = \lambda\Omega^2(Z) + \frac{\rho}{2}\|Z - Q\|_2^2, \qquad \varepsilon_K = \mathcal{F}_\rho(Z^{(K)}) - \mathcal{F}_\rho(Z^*). \tag{3.8}$$

The following proposition translates this objective error into a bound on the proximal iterate error.

Proposition 3.4 (Finite-step proximal error in ADMM)**.** *Since $\mathcal{F}_\rho$ in* (3.8) *is ρ-strongly convex,*

$$\|Z^{(K)} - Z^*\|_2 \le \sqrt{\frac{2\varepsilon_K}{\rho}}. \tag{3.9}$$

If the inner method satisfies $\varepsilon_K = O(K^{-2})$, then

$$\|Z^{(K)} - Z^*\|_2 = O\left(\frac{1}{\sqrt{\rho}K}\right). \tag{3.10}$$

Consequently, Proposition 3.4 shows that a fixed-budget inner solve can leave a nonzero discrepancy $\|Z^{(K)} - Z^*\|_2$, so a block sufficiently close to the zero/nonzero boundary may remain numerically nonzero even when its exact proximal solution is zero. The proof is deferred to Appendix C.3. The effect of this finite-step error on support recovery therefore depends on the zero/nonzero boundary of the exact proximal problem.

3.4 Zero-Block Recovery in ADMM

We answer this question through the KKT condition of the exact proximal problem in (3.7). In particular, the distance of Z^* from the zero/nonzero boundary determines whether the approximation error bounded in Proposition 3.4 can change a block's support status. This connects finite-step proximal accuracy directly to exact zero-block recovery.

Lemma 3.5 (Zero-Block Recovery in ADMM)**.** *For the ADMM proximal problem with exact solution Z^* and proximal input Q, the KKT condition is*

$$0 \in \rho(Z^* - Q) + \lambda\partial\Omega^2(Z^*). \tag{3.11}$$

For a block u, $Z_u^* = 0$ only when ρQ_u lies in the corresponding zero-block subgradient set $\lambda\partial_u\Omega^2(0; Z_{-u}^*)$. Hence, if the finite-step proximal error is smaller than the distance to this zero/nonzero boundary, the approximate and exact proximal updates recover the same support up to numerical tolerance. A useful local scale for this regime is

$$\rho_c^{\mathrm{proxy}} = \frac{d_v}{\|f_v(W)^{p-1}\|_\infty}, \qquad f_v(W) = \left(\sum_{u \in D(v)} \|W_u\|_2^p \right)^{1/p}. \tag{3.12}$$

This result explains why improving proximal accuracy can remove numerical over-selection without changing the underlying HTNL objective. The proof is provided in Appendix C.4.

3.5 ADMM-BISECT: EXACT PROXIMAL UPDATE

Proposition 3.4 shows that a fixed-budget FISTA solve can leave a nonzero proximal residual, while Lemma 3.5 shows that such an error can change the zero/nonzero decision for blocks near the KKT boundary. We therefore replace the fixed-budget FISTA approximation in the ADMM Z-update with a bisection solver (Burden et al., 2015) for the local proximal equations. By solving these equations to numerical tolerance, ADMM-Bisect removes the truncation residual that can cross the KKT boundary and thereby prevents solver-induced retention of blocks whose exact proximal solution is zero.

For the single-geographic-group, scalar-coefficient adapter, let $q = p/(p-1)$ and $c = \lambda d_v/\rho$. The exact local proximal solution satisfies $Z_v^* = 0$ when $\|Q_v\|_q \le c$. Otherwise, when $\|Q_v\|_q > c$, each coordinate has the form

$$Z_{v,u}^* = \operatorname{sign}(Q_{v,u}) r_u, \qquad r_u + \eta r_u^{p-1} = |Q_{v,u}|, \qquad r_u \ge 0. \tag{3.13}$$

The coordinates are coupled through

$$\eta = \frac{c}{(\sum_u r_u^p)^{(p-1)/p}}. \tag{3.14}$$

For fixed $\eta > 0$, the scalar equation in (3.13) is monotone and has a unique root in $[0, |Q_{v,u}|]$, which is solved by bisection. We update η using (3.14) until the local norm is consistent. Since bisection requires $O(\log \varepsilon^{-1})$ iterations to reach tolerance ε, solving $|D_U(v)|$ scalar equations over B_v norm-consistency updates costs $O(B_v |D_U(v)| \log \varepsilon^{-1})$ for each local hierarchical block.

3.6 WEIGHTED GROUP-LASSO REDUCTION

An alternative is to reduce the two-level hierarchy to a sequence of weighted one-level group-lasso problems. Using the dual representation of the outer ℓ_p norm, WGL fixes dual weights for the hierarchical level, solves the resulting weighted group lasso, and then updates those weights in closed form. This connects the HTNL hierarchy to standard group-lasso and regularization-path optimization (Yuan & Lin, 2006; Friedman et al., 2010).

Lemma 3.6 (Weighted one-level reduction)**.** *Let $r_{v,u}(W_U) = \sum_{g\in\mathcal{G}} \|P_{u,g} W_U\|_2$ and let q satisfy $1/p + 1/q = 1$. Then the local hierarchical penalty satisfies*

$$\left(\sum_{u \in D_U(v)} r_{v,u}(W_U)^p \right)^{1/p} = \max_{\alpha_v \ge 0,\ \|\alpha_v\|_q \le 1} \sum_{u \in D_U(v)} \alpha_{v,u} r_{v,u}(W_U). \tag{3.15}$$

For fixed α, the restricted HTNL problem reduces to

$$\min_{W_U} \mathcal{L}(W_U) + \sum_{u\in U} \sum_{g\in\mathcal{G}} c_{u,g}(\alpha) \|P_{u,g} W_U\|_2, \qquad c_{u,g}(\alpha) = \lambda \sum_{v \in A_U(u)} d_v \alpha_{v,u}. \tag{3.16}$$

Thus, fixing the outer dual weights converts the two-level penalty into a weighted one-level group lasso.

WGL alternates the weighted group-lasso solve with closed-form updates of α and applies this procedure along a continuation path. The full reduction and algorithm are given in Appendix B.2.

3.7 ACTIVE-SET ACCESSIBILITY LIMIT

The preceding methods differ in how they solve the restricted HTNL problem, but all share the active-set expansion rule in (2.10). Thus, improving the inner optimization does not guarantee that a predictive NCF can enter the working set. By (2.9), an inactive NCF is admitted only when its KKT score exceeds τ_{KKT}. We characterize this shared accessibility constraint by showing that the admission score decays rapidly with conjunction length.

Let $v \subseteq [P]$ be an NCF of length $k = |v|$, with $\phi_v(X_{s,t}) = \prod_{j\in v} x_j(s,t)$. Assume that $x_j(s,t) \in \{0,1\}$ are independent Bernoulli variables with $\Pr[x_j(s,t) = 1] = p_j \le p_{\max} < 1$. Let $n_{\mathrm{eff}} = |S||T|$ and assume that the loss gradient is uniformly bounded:

$$\|\nabla_W \mathcal{L}(W^*)_{s,t}\|_2 \le G \qquad \text{for all } (s,t). \tag{3.17}$$

Under these assumptions, the KKT admission score in (2.9) admits the following length-dependent bound.

Proposition 3.7 (Exponential decay of active-set KKT scores)**.** *Under the assumptions above, the expected KKT score of a length-k NCF satisfies*

$$\mathbb{E}[\mathrm{score}(v)] \le G n_{\mathrm{eff}} \left(\frac{p_{\max}}{2}\right)^k. \tag{3.18}$$

Consequently, for any threshold $\tau_{\mathrm{KKT}} > 0$, if $k \ge k^* := \left\lceil \frac{\log(G n_{\mathrm{eff}}/\tau_{\mathrm{KKT}})}{\log(2/p_{\max})} \right\rceil$, then $\mathbb{E}[\mathrm{score}(v)] \le \tau_{\mathrm{KKT}}$. Moreover, for any $\epsilon > 0$,

$$\Pr\left[\mathrm{score}(v) \ge \tau_{\mathrm{KKT}} + \epsilon\right] \le \frac{G n_{\mathrm{eff}} \left(p_{\max}/2\right)^k}{\tau_{\mathrm{KKT}} + \epsilon}. \tag{3.19}$$

Equations (3.18)–(3.19) show that both the expected KKT score and the probability of clearing the threshold decay exponentially with NCF length. Thus, Proposition 3.7 establishes an accessibility bias against long conjunctions shared by all solvers, explaining their saturation when useful NCFs cannot enter the active set. The proof is deferred to Appendix C.5.

Corollary 3.8 (Active-set accessibility bound)**.** *Let U^* be the true NCF support, and define the accessible true support as*

$$U^*_{\mathrm{acc}} := \{u \in U^* : |u| \le D \text{ and } \mathbb{E}[\mathrm{score}(u)] > \tau_{\mathrm{KKT}}\}, \tag{3.20}$$

where D is the maximum NCF length allowed by the active-set search. Any optimization using this active-set rule can recover true NCFs only from U^*_{acc} in expectation. Hence the expected true-support recall is bounded by

$$\mathbb{E}\left[\mathrm{Recall}(\hat{S}; U^*)\right] \le \frac{|U^*_{\mathrm{acc}}|}{|U^*|}. \tag{3.21}$$

Together, Proposition 3.7 and Corollary 3.8 show that exponential KKT-score decay restricts recovery to the accessible support U^*_{acc}, imposing the recall ceiling in (3.21). Thus, improving the inner optimization alone cannot recover predictive NCFs excluded by the shared active-set rule. The proof is deferred to Appendix C.6.

4 Controlled Optimization Method Diagnostics

Before turning to real data, we use two controlled synthetic datasets with paired seeds to isolate the mechanisms predicted by our analysis. These diagnostics distinguish algebraic equivalence, finite-step numerical error, and regularization-path effects under controlled conditions.

Table 1 validates Proposition 3.2, with MM and VQF remaining bit-exact under matched auxiliary updates, and tests Proposition 3.4, with exact bisection yielding the only positive F1 improvement that survives Holm–Bonferroni correction. The WGL variants further diagnose regularization-path sensitivity, showing that adaptive penalties do not reliably prevent collapse.

Table 1: NCF support-selection F1 on two controlled synthetic datasets. Results are mean ± standard deviation over 20 paired random seeds at active-set thresholds $\tau = 0.20$ and 0.40.

	Dataset 1		**Dataset 2**	
Method	$\tau = 0.20$	$\tau = 0.40$	$\tau = 0.20$	$\tau = 0.40$
HTNL-MM	0.580 ± 0.201	0.592 ± 0.253	0.581 ± 0.188	0.590 ± 0.246
HTNL-VQF	0.580 ± 0.201	0.592 ± 0.253	0.581 ± 0.188	0.590 ± 0.246
HTNL-ADMM (FISTA, $\rho = 10$)	0.468 ± 0.133	$\mathbf{0.674 \pm 0.187}$	0.457 ± 0.147	$\mathbf{0.674 \pm 0.211}$
HTNL-WGL ($\lambda = 100$)	0.463 ± 0.336	0.463 ± 0.336	0.360 ± 0.306	0.360 ± 0.306
Best-of-class variants (post-hoc sweeps)				
ADMM-Bisect ($\rho = 1$, exact prox)	0.488 ± 0.128	0.685 ± 0.192	0.468 ± 0.137	0.687 ± 0.213
WGL-Cont (high-to-low, $500 \to 50$)	0.602 ± 0.250	0.549 ± 0.273	0.607 ± 0.231	0.567 ± 0.273
WGL-Adaptive (0.2–5.0)	0.403 ± 0.195	0.403 ± 0.195	0.383 ± 0.214	0.383 ± 0.214

Figures 4 and 2b isolate the ADMM mechanism. Consistent with Proposition 3.4, exact bisection never underperforms finite-step FISTA across paired seeds in the significant comparison. Combined with KKT pruning, it preserves F1 while reducing runtime by excluding low-violation blocks before the inner solve.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

We evaluate the HTNL optimization methods on **11 real-world binary classification datasets** (Zhao et al., 2014; Compton et al., 2014b): 10 country-level Latin American Twitter civil-unrest corpora and one US flu-surveillance dataset. The civil-unrest datasets contain $V{=}100$ primitive feature groups and exhibit severe class imbalance, with positive rates of 0.5–1.5%. The flu dataset contains $V{=}182$ feature groups, $n_{\text{train}}{=}5{,}096$, and a positive rate of 7.4%. These datasets span severely imbalanced event forecasting and a comparatively less imbalanced public-health prediction setting.

We use held-out AUC as the primary evaluation metric because it provides a threshold-independent assessment of ranking performance (Hanley & McNeil, 1982; Fawcett, 2006). Given the severe class imbalance, we additionally report precision–recall-based metrics where appropriate, which complement ROC-based evaluation in highly skewed settings (Davis & Goadrich, 2006). All HTNL variants use the same class-weighted Huber-hinge objective (Huber, 1964) without resampling. Statistical comparisons against Logistic Regression use paired t-tests across the 11 datasets, with Bonferroni correction (Bonferroni, 1936) over four pre-specified HTNL comparisons.

Unless stated otherwise, all methods use the same active-set configuration. WGL follows a four-step continuation schedule from $\lambda_{\text{init}}{=}500$ to $\lambda_{\text{target}}{=}50$; we additionally evaluate a scale-corrected $\lambda{=}1$ variant motivated by the class-imbalance analysis in Section D.1.

5.2 PREDICTIVE PERFORMANCE ON REAL-WORLD DATASETS

We compare the HTNL variants with class-weighted Logistic Regression on identical train/test splits (Table 2 and Figure 6). All non-collapsing HTNL variants improve over the baseline after Bonferroni correction (Bonferroni, 1936). VQF performs best, increasing mean AUC by 0.074 and outperforming Logistic Regression on all 11 datasets. ADMM-Bisect and WGL($\lambda = 1$) also improve predictive performance, but their broader active sets provide no additional benefit over the sparser VQF solution. This suggests that broader feature selection does not necessarily translate into better generalization. Complementary synthetic diagnostics further show that ADMM is more sensitive than VQF to the post-hoc NCF selection threshold, with their relative F1 ranking changing across thresholds (Appendix 5.4).

WGL further reveals the importance of optimization scale. Its default continuation schedule collapses to an empty model with AUC $= 0.500$ on all 10 highly imbalanced civil-unrest datasets, whereas the scale-corrected WGL($\lambda = 1$) recovers non-degenerate solutions and predictive gains. Appendix D.1 provides the full regularization-scale analysis, while Appendix F.5 further examines this behavior across different positive-class fractions.

Table 2: Predictive AUC on 11 real-world datasets using identical train/test splits. n and pos% denote the number of training instances and positive-class fraction, respectively. ΔAUC denotes the improvement of VQF over Logistic Regression. The final row reports mean $\pm$ standard deviation across datasets.

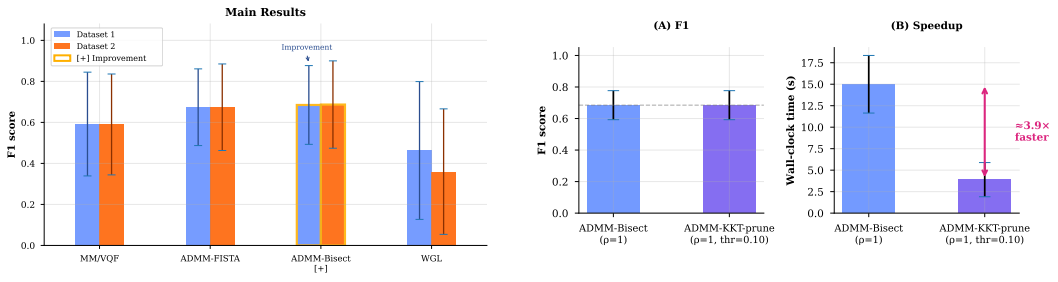
Dataset	n	pos%	**LogReg**	**MM/VQF**	**ADMM-Bisect**	**WGL($\lambda = 1$)**	ΔAUC
Flu$\star$	5,096	7.40%	0.649	0.695	0.600	0.606	+0.046
Ecuador	5,916	0.74%	0.529	0.604	0.563	0.581	+0.075
El Salvador	8,700	1.26%	0.477	0.572	0.559	0.555	+0.095
Uruguay	10,788	0.82%	0.481	0.601	0.565	0.582	+0.120
Venezuela	11,832	0.70%	0.463	0.524	0.514	0.523	+0.061
Chile	12,876	0.49%	0.459	0.527	0.501	0.503	+0.068
Paraguay	14,964	1.52%	0.518	0.564	0.549	0.557	+0.047
Colombia	19,140	0.51%	0.523	0.566	0.529	0.547	+0.043
Argentina	19,836	0.51%	0.453	0.604	0.494	0.517	+0.152
Brazil	22,620	0.48%	0.531	0.598	0.584	0.582	+0.067
Mexico	41,760	1.29%	0.590	0.629	0.595	0.602	+0.039
Grand mean	—	—	0.516 ± 0.060	0.589 ± 0.048	0.550 ± 0.037	0.559 ± 0.035	+0.074

Figure 2a summarizes the synthetic results.MM and VQF remain identical, while replacing FISTA with bisection improves ADMM F1 (ΔF1 $= +0.088$, $p_{\text{adj}} = 2.60 \times 10^{-3}$). In contrast, WGL-Adaptive significantly degrades performance on both datasets. Per-dataset comparisons and runtime

results are provided in Appendix F.4; MM/VQF achieves the highest AUC among the evaluated HTNL methods on all 10 LatAm datasets.

5.3 KKT Screening and Computational Efficiency

We further evaluate the computational effect of combining KKT pruning with the bisection proximal solver. Figure 2b compares the baseline ADMM configuration at $\rho = 1$ with ADMM using KKT pruning and bisection at the same ρ, with pruning threshold 0.10. The two configurations achieve nearly identical synthetic F1, while the KKT-pruned bisection variant substantially reduces wall-clock time. In this diagnostic, runtime decreases from approximately 15 s to 4 s, corresponding to an approximately $3.9\times$ speedup. This result shows that the combined pruning and bisection implementation can reduce computational cost without an observable loss in F1 under this experimental condition.



(a) Synthetic F1. (b) KKT pruning: F1 and runtime.

Figure 2: Controlled optimization diagnostics. **Left:** Bisection improves ADMM F1, while MM and VQF are identical. **Right:** KKT pruning preserves F1 while reducing runtime.

5.4 Sensitivity to the Active-Set Admission Threshold

Figure 3 examines sensitivity to the active-set admission threshold τ_{KKT} in Equation 2.10, which determines whether an inactive NCF is admitted to the active set based on its KKT score. ADMM is more sensitive to this threshold than VQF: changing τ_{KKT} alters the admitted candidates and can reverse their relative F1 ranking, whereas VQF remains comparatively stable. This result shows that optimization comparisons can depend on the active-set admission threshold and motivates evaluation across multiple threshold settings.

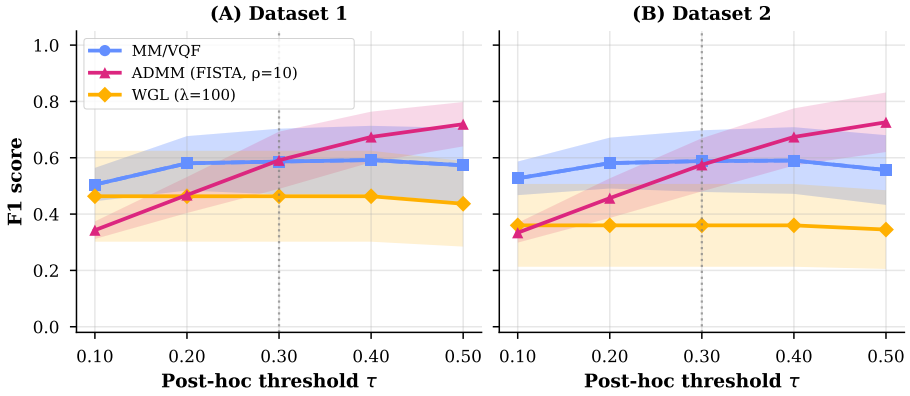


Figure 3: Synthetic F1 across active-set admission thresholds τ_{KKT}. Shaded bands show uncertainty across paired random seeds.

6 Conclusion

We studied four HTNL optimization methods for Numerical Conjunctive Feature selection. Our analysis shows that MM and VQF are equivalent under matched updates, ADMM-Bisect corrects the inexact proximal step, and all methods remain limited by the shared active-set KKT rule. Across 11 real-world datasets, non-collapsing HTNL variants improve over Logistic Regression, while WGL requires careful regularization scaling. Overall, improving the inner solver helps optimization, but overcoming the remaining active-set limit requires a better candidate-expansion strategy.

AI USE STATEMENT

AI tools assisted with LaTeX formatting, organization of the main text and appendices, and layout checks. The original theoretical statements and reported experimental were retained.

REFERENCES

Harshavardhan Achrekar, Avinash Gandhe, Ross Lazarus, Ssu-Hsin Yu, and Benyuan Liu. Predicting flu trends using Twitter data. In *IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 702–707, 2011.

Andreas Argyriou, Theodoros Evgeniou, and Massimiliano Pontil. Convex multi-task feature learning. *Machine Learning*, 73(3):243–272, 2008.

Francis Bach. High-dimensional non-linear variable selection through hierarchical kernel learning. Technical Report arXiv:0909.0844, arXiv, 2009.

Francis Bach, Rodolphe Jenatton, Julien Mairal, and Guillaume Obozinski. Optimization with sparsity-inducing penalties. *Foundations and Trends in Machine Learning*, 4(1):1–106, 2012. doi: 10.1561/2200000015.

Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

Carlo E. Bonferroni. Teoria statistica delle classi e calcolo delle probabilità. *Pubblicazioni del R Istituto Superiore di Scienze Economiche e Commerciali di Firenze*, 8:3–62, 1936.

Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. *Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers*. Now Publishers, 2011.

Richard L. Burden, J. Douglas Faires, and Annette M. Burden. *Numerical Analysis*. Cengage Learning, 10th edition, 2015.

Jose Cadena, Gizem Korkmaz, Chris J. Kuhlman, Achla Marathe, Naren Ramakrishnan, and Anil Vullikanti. Forecasting social unrest using activity cascades. *PLOS ONE*, 10(6):e0128879, 2015. doi: 10.1371/journal.pone.0128879.

Hong Cheng, Xifeng Yan, Jiawei Han, and Chih-Wei Hsu. Discriminative frequent pattern analysis for effective classification. In *Proceedings of the 23rd IEEE International Conference on Data Engineering (ICDE)*, pp. 716–725, 2007.

Tanmoy Chowdhury, Yuyang Gao, and Liang Zhao. Deep multi-task learning for spatio-temporal incomplete qualitative event forecasting. *IEEE Transactions on Knowledge and Data Engineering*, 36(12):7913–7926, 2024.

Ryan Compton, Craig Lee, Jiejun Xu, Luis Artieda-Moncada, Tsai-Ching Lu, Lalindra De Silva, and Michael Macy. Using publicly visible social media to build detailed forecasts of civil unrest, 2014a.

Ryan Compton, Craig Lee, Jiejun Xu, Luis Artieda-Moncada, Tsai-Ching Lu, Lalindra De Silva, and Michael Macy. Using publicly visible social media to build detailed forecasts of civil unrest. *Security Informatics*, 3(1):4, 2014b. doi: 10.1186/s13388-014-0004-6.

Gao Cong, Kian-Lee Tan, Anthony K. H. Tung, and Xin Xu. Mining top-k covering rule groups for gene expression data. In *Proceedings of the ACM SIGMOD*, pp. 670–681, 2005.

Jesse Davis and Mark Goadrich. The relationship between precision-recall and ROC curves. In *Proceedings of the 23rd International Conference on Machine Learning*, pp. 233–240, 2006. doi: 10.1145/1143844.1143874.

Songgaojun Deng and Yue Ning. A survey on societal event forecasting with deep learning. *arXiv preprint*, 2021. arXiv preprint arXiv:2112.06345.

Songgaojun Deng, Maarten de Rijke, and Yue Ning. Advances in human event modeling: From graph neural networks to language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 6527–6537, 2024. doi: 10.1145/3637528.3671466.

Jonathan Eckstein and Dimitri P. Bertsekas. On the Douglas–Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Mathematical Programming*, 55(1–3): 293–318, 1992. doi: 10.1007/BF01581204.

Tom Fawcett. An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8):861–874, 2006. doi: 10.1016/j.patrec.2005.10.010.

Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22, 2010. doi: 10.18637/jss.v033.i01.

Jerome H. Friedman and Bogdan E. Popescu. Predictive learning via rule ensembles. *The Annals of Applied Statistics*, 2(3):916–954, 2008.

Matthew S. Gerber. Predicting crime using Twitter and kernel density estimation. *Decision Support Systems*, 61:115–125, 2014.

Mehmet Gönen and Ethem Alpaydın. Multiple kernel learning algorithms. *Journal of Machine Learning Research*, 12:2211–2268, 2011.

Brian J. Goode, Siddharth Krishnan, Michael Roan, and Naren Ramakrishnan. Pricing a protest: Forecasting the dynamics of civil unrest activity in social media. *PLOS ONE*, 10(10):e0139911, 2015. doi: 10.1371/journal.pone.0139911.

James A. Hanley and Barbara J. McNeil. The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*, 143(1):29–36, 1982.

Godfrey H. Hardy, John E. Littlewood, and George Pólya. *Inequalities*. Cambridge University Press, Cambridge, 2 edition, 1952.

Peter J. Huber. Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, 35 (1):73–101, 1964. doi: 10.1214/aoms/1177703732.

David R. Hunter and Kenneth Lange. A tutorial on MM algorithms. *The American Statistician*, 58 (1):30–37, 2004. doi: 10.1198/0003130042836.

Laurent Jacob, Guillaume Obozinski, and Jean-Philippe Vert. Group lasso with overlap and graph lasso. In *Proceedings of the 26th International Conference on Machine Learning (ICML)*, pp. 433–440, 2009.

Rodolphe Jenatton, Jean-Yves Audibert, and Francis Bach. Structured variable selection with sparsity-inducing norms, 2011.

Seyoung Kim and Eric P. Xing. Tree-guided group lasso for multi-task regression with structured sparsity. In *Proceedings of the 27th International Conference on Machine Learning (ICML)*, pp. 543–550, 2010.

Marius Kloft, Ulf Brefeld, Sören Sonnenburg, and Alexander Zien. ℓ_p-norm multiple kernel learning. *Journal of Machine Learning Research*, 12:953–997, 2011.

Alex Lamb, Michael J. Paul, and Mark Dredze. Separating fact from fear: Tracking flu infections on Twitter. In *Proceedings of HLT-NAACL*, pp. 789–795, 2013.

Haoxuan Li, Zhengmao Yang, Yunshan Ma, Yi Bin, Yang Yang, and Tat-Seng Chua. MM-Forecast: A multimodal approach to temporal event forecasting with large language models. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pp. 2776–2785, 2024. doi: 10.1145/3664647.3681593. URL https://arxiv.org/abs/2408.04388.

Ruotong Liao, Xu Jia, Yangzhe Li, Yunpu Ma, and Volker Tresp. GenTKG: Generative forecasting on temporal knowledge graph with large language models. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pp. 4303–4317, Mexico City, Mexico, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-naacl.268. URL https://aclanthology.org/2024.findings-naacl.268/.

Lukas Meier, Sara Van de Geer, and Peter Bühlmann. The group lasso for logistic regression. *Journal of the Royal Statistical Society, Series B*, 70(1):53–71, 2008.

Yue Ning, Rongrong Tao, Chandan K. Reddy, Huzefa Rangwala, James C. Starz, and Naren Ramakrishnan. STAPLE: Spatio-temporal precursor learning for event forecasting. In *Proceedings of the 2018 SIAM International Conference on Data Mining*, pp. 99–107, 2018. doi: 10.1137/1.9781611975321.17.

Guillaume Obozinski, Ben Taskar, and Michael Jordan. Joint covariate selection and joint subspace selection for multiple classification problems. *Statistics and Computing*, 20(2):231–252, 2010.

Michael R. Osborne, Brett Presnell, and Berwin A. Turlach. On the LASSO and its dual. *Journal of Computational and Graphical Statistics*, 9(2):319–337, 2000.

Neal Parikh and Stephen Boyd. Proximal algorithms. *Foundations and Trends in Optimization*, 1(3): 127–239, 2014.

Naren Ramakrishnan, Patrick Butler, et al. Beating the news with EMBERS: Forecasting civil unrest using open source indicators. In *Proceedings of the 20th ACM SIGKDD*, pp. 1799–1808, 2014.

Ulrich Rückert and Stefan Kramer. A statistical approach to rule learning. In *Proceedings of the 23rd International Conference on Machine Learning (ICML)*, pp. 785–792, 2006.

Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. Earthquake shakes Twitter users: Real-time event detection by social sensors. In *Proceedings of the 19th International Conference on the World Wide Web (WWW)*, pp. 851–860, 2010.

Noah Simon, Jerome Friedman, Trevor Hastie, and Robert Tibshirani. A sparse-group lasso. *Journal of Computational and Graphical Statistics*, 22(2):231–245, 2013.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58(1):267–288, 1996.

Robert Tibshirani, Jacob Bien, Jerome Friedman, Trevor Hastie, Noah Simon, Jonathan Taylor, and Ryan J. Tibshirani. Strong rules for discarding predictors in lasso-type problems. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 74(2):245–266, 2012. doi: 10.1111/j.1467-9868.2011.01004.x.

Ryan J. Tibshirani and Jonathan Taylor. Degrees of freedom in lasso problems. *Annals of Statistics*, 40(2):1198–1232, 2012.

Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society, Series B*, 68(1):49–67, 2006.

Liang Zhao, Feng Chen, Jing Dai, Ting Hua, Chang-Tien Lu, and Naren Ramakrishnan. Unsupervised spatial event detection in targeted domains with applications to civil unrest modeling. *PLoS ONE*, 9(10):e110206, 2014. doi: 10.1371/journal.pone.0110206.

Liang Zhao, Feng Chen, Chang-Tien Lu, and Naren Ramakrishnan. Spatiotemporal event forecasting in social media. In *Proceedings of the SIAM International Conference on Data Mining (SDM)*, pp. 963–971, 2015.

Liang Zhao, Feng Chen, Chang-Tien Lu, and Naren Ramakrishnan. Multi-resolution spatial event forecasting in social media. In *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, pp. 689–698, 2016.

Liang Zhao, Feng Chen, and Yanfang Ye. Efficient learning with exponentially-many conjunctive precursors to forecast spatial events. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 2019. doi: 10.1109/TKDE.2019.2912187.

A Related Work

Spatial event forecasting and conjunctive precursor learning. Forecasting societal events from social media has been studied in civil unrest, disease outbreaks, crime prediction, and other spatial-temporal settings (Ramakrishnan et al., 2014; Gerber, 2014; Compton et al., 2014a; Lamb et al., 2013; Achrekar et al., 2011). Many systems use keyword counts, topic volumes, burstiness, or spatial density signals as predictive features. These features are scalable, but they mostly model single-factor evidence. NCFs instead represent interactions among primitive signals, making them closer to rule-learning and precursor-discovery methods (Friedman & Popescu, 2008; Rückert & Kramer, 2006; Cheng et al., 2007; Cong et al., 2005). HTNL (Zhao et al., 2019) extends this direction by learning numerical conjunctions directly from an exponential lattice while also modeling geographic structure.

Structured sparsity and hierarchical selection. HTNL belongs to a broad family of sparsity-inducing methods, including lasso (Tibshirani, 1996), group lasso (Yuan & Lin, 2006; Meier et al., 2008), overlapping group lasso (Jacob et al., 2009), sparse-group lasso (Simon et al., 2013), hierarchical sparsity (Bach, 2009; Jenatton et al., 2011; Bach et al., 2012), and multi-task feature selection (Argyriou et al., 2008; Obozinski et al., 2010; Kim & Xing, 2010). These methods provide the modeling language for group and hierarchy selection, but HTNL adds two complications that are central to this paper: an inner geographic group norm nested inside an outer NCF hierarchy, and an active-set search over exponentially many candidate conjunctions. Multiple kernel learning is also related through its use of norm-controlled component selection (Gönen & Alpaydın, 2011; Kloft et al., 2011), but it does not address active-set NCF recovery.

Proximal optimization, ADMM, and active sets. The HTNL objective combines a smooth convex loss with a nonsmooth hierarchical regularizer, making it closely related to proximal algorithms, splitting methods, and working-set optimization (Beck & Teboulle, 2009; Boyd et al., 2011; Eckstein & Bertsekas, 1992; Parikh & Boyd, 2014; Bach et al., 2012). ADMM is a standard tool for such composite objectives, but support recovery can be sensitive to inexact inner proximal solves when sparsity is the target. Active-set and path methods for lasso-style problems reduce computation by adding variables that violate optimality conditions (Osborne et al., 2000; Tibshirani & Taylor, 2012; Friedman et al., 2010). Our contribution is to diagnose how this active-set mechanism behaves in the NCF lattice: improving the inner solver can remove numerical artifacts, but the KKT admission rule itself can still limit which high-order conjunctions are reachable.

Neural and language-model event forecasting. Recent event forecasting work increasingly uses deep spatial-temporal representations, graph neural networks, and language models (Deng & Ning, 2021; Deng et al., 2024; Chowdhury et al., 2024; Liao et al., 2024; Li et al., 2024). These methods are powerful predictors, but they usually return latent representations, generated events, or end-task predictions rather than sparse, inspectable precursor rules. HTNL is complementary: it targets transparent NCF discovery, and our paper focuses on making the structured-sparsity optimization reliable enough to support that interpretability goal.

B Detailed Optimization Method Formulations

B.1 Direct ADMM optimization on a fixed active set

The variational quadratic form in Section 3.1 converts the HTNL regularizer into a weighted ridge problem after auxiliary variables are fixed. We next consider a direct ADMM solver that keeps the original nonsmooth two-level group lasso structure. The key idea is to split each active NCF block into auxiliary consensus variables, so that the smooth loss is optimized in the W-update and the nested group penalty is handled in the Z-update.

Let $U \subseteq \mathcal{V}$ be the current active set of numerical conjunctive features (NCFs). We write the restricted parameter collection as

$$W_U = \{W_{j,u} : u \in U,\ j \in S\}, \tag{B.1}$$

where u indexes an NCF, j indexes a spatial location, and S denotes the full set of spatial locations. Let $\mathcal{G}$ be the set of geographical groups, where each $g \in \mathcal{G}$ is a subset of S. For each $u \in U$ and

$g \in \mathcal{G}$, define the block extraction operator

$$P_{u,g} W_U = \{W_{j,u} : j \in g\}. \tag{B.2}$$

Thus, $P_{u,g} W_U$ extracts the parameter block associated with NCF u and geographical group g.

On the active set U, define the restricted descendant set

$$D_U(v) = D(v) \cap U. \tag{B.3}$$

The direct nonsmooth optimization problem is

$$\min_{W_U} \mathcal{L}(W_U) + \lambda \sum_{v \in U} d_v \left(\sum_{u \in D_U(v)} \left(\sum_{g \in \mathcal{G}} \|P_{u,g} W_U\|_2 \right)^p \right)^{1/p}. \tag{B.4}$$

The difficulty in Equation (B.4) is that the regularizer contains both the geographical group norm $\sum_{g \in \mathcal{G}} \|P_{u,g} W_U\|_2$ and the hierarchical coupling over $u \in D_U(v)$. Therefore, a direct proximal step over W_U is not generally available in closed form.

To apply ADMM, we introduce one auxiliary copy for every block appearing in every hierarchical regularization term:

$$Z_{v,u,g} = P_{u,g} W_U, \qquad v \in U,\ u \in D_U(v),\ g \in \mathcal{G}. \tag{B.5}$$

The constrained reformulation of Equation (B.4) is

$$\begin{aligned} &\min_{W_U, Z} \mathcal{L}(W_U) + \lambda \sum_{v \in U} d_v \left(\sum_{u \in D_U(v)} \left(\sum_{g \in \mathcal{G}} \|Z_{v,u,g}\|_2 \right)^p \right)^{1/p} \\ &\text{s.t.} \quad Z_{v,u,g} = P_{u,g} W_U,\ v \in U,\ u \in D_U(v),\ g \in \mathcal{G}. \end{aligned} \tag{B.6}$$

This splitting places the smooth loss on W_U and the nonsmooth double group regularizer on Z, with the two parts connected only through linear consensus constraints.

Using scaled dual variables $Y_{v,u,g}$ and penalty parameter $\rho > 0$, the scaled augmented Lagrangian is

$$\begin{aligned} \mathcal{L}_\rho(W_U, Z, Y) = \mathcal{L}(W_U) &+ \lambda \sum_{v \in U} d_v \left(\sum_{u \in D_U(v)} \left(\sum_{g \in \mathcal{G}} \|Z_{v,u,g}\|_2 \right)^p \right)^{1/p} \\ &+ \frac{\rho}{2} \sum_{v \in U} \sum_{u \in D_U(v)} \sum_{g \in \mathcal{G}} \|P_{u,g} W_U - Z_{v,u,g} + Y_{v,u,g}\|_2^2. \end{aligned} \tag{B.7}$$

ADMM then alternates the following three updates. First, the W_U-update solves

$$W_U^{k+1} = \arg\min_{W_U} \mathcal{L}(W_U) + \frac{\rho}{2} \sum_{v \in U} \sum_{u \in D_U(v)} \sum_{g \in \mathcal{G}} \|P_{u,g} W_U - Z_{v,u,g}^k + Y_{v,u,g}^k\|_2^2. \tag{B.8}$$

This subproblem contains no nonsmooth group norm. If $\mathcal{L}$ is squared loss, Equation (B.8) is a quadratic least-squares problem. If $\mathcal{L}$ is logistic or another smooth convex loss, Equation (B.8) can be solved by Newton, L-BFGS, or an inner gradient method. The quadratic term in Equation (B.8) is an ADMM consensus penalty, not a variational ridge reformulation of the original regularizer.

Second, define

$$Q_{v,u,g}^{k+1} = P_{u,g} W_U^{k+1} + Y_{v,u,g}^k. \tag{B.9}$$

The Z-update is

$$\begin{aligned} Z^{k+1} = \arg\min_{Z} \lambda &\sum_{v \in U} d_v \left(\sum_{u \in D_U(v)} \left(\sum_{g \in \mathcal{G}} \|Z_{v,u,g}\|_2 \right)^p \right)^{1/p} \\ &+ \frac{\rho}{2} \sum_{v \in U} \sum_{u \in D_U(v)} \sum_{g \in \mathcal{G}} \|Z_{v,u,g} - Q_{v,u,g}^{k+1}\|_2^2. \end{aligned} \tag{B.10}$$

Because the regularizer is separable over the ancestor index v, Equation (B.10) decomposes into independent subproblems:

$$Z_v^{k+1} = \arg\min_{Z_v} \lambda d_v \left(\sum_{u \in D_U(v)} \left(\sum_{g \in \mathcal{G}} \|Z_{v,u,g}\|_2 \right)^p \right)^{1/p} + \frac{\rho}{2} \sum_{u \in D_U(v)} \sum_{g \in \mathcal{G}} \|Z_{v,u,g} - Q_{v,u,g}^{k+1}\|_2^2. \tag{B.11}$$

Equivalently, define

$$\Omega_v(Z_v) = \left(\sum_{u \in D_U(v)} \left(\sum_{g \in \mathcal{G}} \|Z_{v,u,g}\|_2 \right)^p \right)^{1/p}. \tag{B.12}$$

Then Equation (B.11) can be written as the proximal update

$$Z_v^{k+1} = \mathrm{prox}_{(\lambda d_v/\rho)\Omega_v}(Q_v^{k+1}). \tag{B.13}$$

This proximal operator is not the ordinary group soft-thresholding operator, because Ω_v contains two coupled levels: an inner sum of geographical ℓ_2 group norms over $g \in \mathcal{G}$, followed by an outer ℓ_p aggregation over $u \in D_U(v)$. Thus, unless the special choice of p and group structure gives a closed form, Equation (B.13) must itself be solved by an inner proximal, dual-projection, or primal-dual splitting routine.

Third, the scaled dual variables are updated by

$$Y_{v,u,g}^{k+1} = Y_{v,u,g}^k + P_{u,g} W_U^{k+1} - Z_{v,u,g}^{k+1}. \tag{B.14}$$

This update accumulates the violation of the consensus constraint in Equation (B.5). The primal residual for the consensus constraint is

$$r_{v,u,g}^{k+1} = P_{u,g} W_U^{k+1} - Z_{v,u,g}^{k+1}. \tag{B.15}$$

The corresponding dual residual is

$$s^{k+1} = \rho \sum_{v \in U} \sum_{u \in D_U(v)} \sum_{g \in \mathcal{G}} P_{u,g}^\top \left(Z_{v,u,g}^{k+1} - Z_{v,u,g}^k \right), \tag{B.16}$$

where $P_{u,g}^\top$ is the adjoint block insertion operator, which places a block back into the coordinates of W_U. The ADMM inner loop is terminated when

$$\|r^{k+1}\|_2 \le \epsilon_{\mathrm{pri}}, \qquad \|s^{k+1}\|_2 \le \epsilon_{\mathrm{dual}}. \tag{B.17}$$

ADMM directly optimizes the original nonsmooth double group lasso on a fixed active set U by splitting each block $P_{u,g} W_U$ into copies $Z_{v,u,g}$. The W_U-update in Equation (B.8) solves a smooth loss plus a quadratic consensus penalty, while the Z-update in Equation (B.10) keeps the original double group lasso structure and computes its proximal map. For the full HTNL problem, this ADMM routine should still be embedded inside an outer active-set procedure: after solving Equation (B.4) on U, one checks the optimality violation over inactive NCFs in the lattice, adds violating descendants to U, and repeats until no inactive NCF violates the optimality condition.

B.2 Reduction from two-level group lasso to weighted one-level group lasso

The ADMM formulation in Section B.1 preserves the original two-level nonsmooth structure but requires solving a nested proximal problem. We now present an alternative reduction that converts the local ℓ_p hierarchy into a weighted one-level group lasso problem. The main idea is to use the dual representation of the outer ℓ_p norm: after fixing the dual weights, the two-level penalty becomes an ordinary weighted group-lasso penalty over geographic blocks. The hierarchy is restored by updating the dual weights in closed form.

We use the same fixed active set U, parameter collection W_U, block extraction operator $P_{u,g}$, and restricted descendant set $D_U(v)$ defined in (B.1), (B.2), (B.3). Under this notation, the restricted two-level group lasso problem is the same as Equation (B.4). Define the inner geographical group magnitude

$$r_{v,u}(W_U) = \sum_{g \in \mathcal{G}} \|P_{u,g} W_U\|_2, \qquad v \in U,\ u \in D_U(v). \tag{B.18}$$

Then the local hierarchical penalty associated with ancestor v is

$$\Omega_v(W_U) = \left(\sum_{u \in D_U(v)} r_{v,u}(W_U)^p \right)^{1/p}. \tag{B.19}$$

Let q be the dual exponent of p:

$$\frac{1}{p} + \frac{1}{q} = 1. \tag{B.20}$$

By the dual representation of the ℓ_p norm, Equation (B.19) can be written as

$$\Omega_v(W_U) = \max_{\alpha_v \in \mathcal{A}_v} \sum_{u \in D_U(v)} \alpha_{v,u} r_{v,u}(W_U), \tag{B.21}$$

where

$$\mathcal{A}_v = \left\{ \alpha_v \in \mathbb{R}_+^{|D_U(v)|} : \|\alpha_v\|_q \le 1 \right\}. \tag{B.22}$$

Substituting Equation (B.21) into Equation (B.4) gives the equivalent saddle-point formulation

$$\min_{W_U} \max_{\{\alpha_v \in \mathcal{A}_v\}_{v \in U}} \mathcal{L}(W_U) + \lambda \sum_{v \in U} d_v \sum_{u \in D_U(v)} \alpha_{v,u} \sum_{g \in \mathcal{G}} \|P_{u,g} W_U\|_2. \tag{B.23}$$

For fixed $\alpha = \{\alpha_v\}_{v \in U}$, collect the coefficients that multiply the same block norm $\|P_{u,g} W_U\|_2$. Define the active ancestor set

$$A_U(u) = \{v \in U : u \in D_U(v)\}. \tag{B.24}$$

The collapsed block weight is

$$c_{u,g}(\alpha) = \lambda \sum_{v \in A_U(u)} d_v \alpha_{v,u}. \tag{B.25}$$

Since $c_{u,g}(\alpha)$ does not depend on g under the penalty in Equation (B.4), one may also write $c_u(\alpha) = \lambda \sum_{v \in A_U(u)} d_v \alpha_{v,u}$. We keep the notation $c_{u,g}(\alpha)$ to match the blockwise group-lasso form. For fixed α, Equation (B.23) reduces to the weighted one-level group lasso problem

$$\min_{W_U} \mathcal{L}(W_U) + \sum_{u \in U} \sum_{g \in \mathcal{G}} c_{u,g}(\alpha) \|P_{u,g} W_U\|_2. \tag{B.26}$$

One-level group-lasso update. Assume that $\mathcal{L}$ is differentiable and has $L_{\mathcal{L}}$-Lipschitz gradient. Given a stepsize $\eta \le 1/L_{\mathcal{L}}$, the proximal-gradient update for Equation (B.26) is

$$\widetilde{W}_U^k = W_U^k - \eta \nabla \mathcal{L}(W_U^k). \tag{B.27}$$

Then

$$W_U^{k+1} = \mathrm{prox}_{\eta \sum_{u \in U} \sum_{g \in \mathcal{G}} c_{u,g}(\alpha) \|P_{u,g} \cdot\|_2}(\widetilde{W}_U^k). \tag{B.28}$$

If the groups $\{P_{u,g}\}_{u,g}$ are non-overlapping, Equation (B.28) has the block soft-thresholding solution

$$P_{u,g} W_U^{k+1} = \left(1 - \frac{\eta c_{u,g}(\alpha)}{\|P_{u,g} \widetilde{W}_U^k\|_2} \right)_+ P_{u,g} \widetilde{W}_U^k. \tag{B.29}$$

If the geographical groups overlap, Equation (B.26) remains a one-level overlapping group lasso problem and can be solved by a standard overlapping-group solver, such as latent group lasso, ADMM, or a dual projected method.

Outer dual-weight update. Given W_U, the maximizer in Equation (B.21) has a closed form when $1 < p < \infty$. For each $v \in U$ and $u \in D_U(v)$,

$$\alpha_{v,u}^\star(W_U) = \frac{r_{v,u}(W_U)^{p-1}}{\left(\sum_{u' \in D_U(v)} r_{v,u'}(W_U)^p \right)^{(p-1)/p}}. \tag{B.30}$$

If $r_{v,u}(W_U) = 0$ for all $u \in D_U(v)$, any $\alpha_v \in \mathcal{A}_v$ is optimal. In implementation, one can replace $r_{v,u}(W_U)$ by $r_{v,u}(W_U) + \epsilon$ for a small $\epsilon > 0$.

Algorithm 1: Two-Level Group Lasso via Weighted One-Level Group Lasso

Require: Active set U, loss $\mathcal{L}$, groups $\mathcal{G}$, descendant sets $\{D_U(v)\}_{v\in U}$, weights $\{d_v\}_{v\in U}$, regularization parameter λ, exponent p, tolerance ε.
Ensure: Optimized parameter W_U.
1: Initialize W_U^0 and $\alpha^0 = \{\alpha_v^0\}_{v\in U}$ with $\alpha_v^0 \in \mathcal{A}_v$.
2: **for** $t = 0, 1, 2, \ldots$ **do**
3: Compute collapsed one-level group weights

$$c_{u,g}^t = \lambda \sum_{v\in A_U(u)} d_v \alpha_{v,u}^t, \qquad u \in U,\ g \in \mathcal{G}. \tag{B.31}$$

4: Solve the weighted one-level group lasso subproblem

$$W_U^{t+1} = \arg\min_{W_U} \mathcal{L}(W_U) + \sum_{u\in U}\sum_{g\in\mathcal{G}} c_{u,g}^t \|P_{u,g} W_U\|_2. \tag{B.32}$$

5: Compute inner group magnitudes

$$r_{v,u}^{t+1} = \sum_{g\in\mathcal{G}} \|P_{u,g} W_U^{t+1}\|_2, \qquad v \in U,\ u \in D_U(v). \tag{B.33}$$

6: Update outer dual weights

$$\alpha_{v,u}^{t+1} = \frac{(r_{v,u}^{t+1} + \epsilon)^{p-1}}{\left(\sum_{u'\in D_U(v)} (r_{v,u'}^{t+1} + \epsilon)^p\right)^{(p-1)/p}}, \qquad v \in U,\ u \in D_U(v). \tag{B.34}$$

7: **if** $\|W_U^{t+1} - W_U^t\|_F \le \varepsilon$ **then**
8: **break**
9: **end if**
10: **end for**
11: **return** W_U^{t+1}.

The transformation in Equation (B.23) is exact only when the outer dual variables α are optimized together with W_U. For fixed α, the W_U-subproblem in Equation (B.26) is a weighted one-level group lasso problem. The update in Equation (B.30) restores the original outer ℓ_p hierarchy by choosing the dual weights that exactly represent the local ℓ_p norm in Equation (B.19). Therefore, the full procedure alternates between a standard one-level group-lasso solve and a closed-form update of the outer hierarchy weights.

The reduction above is performed on a fixed active set U. It does not remove the exponential number of candidate NCFs in the full feature conjunction lattice. Therefore, the method should be embedded inside an outer active-set procedure: solve the restricted problem on U using Algorithm 1, check the optimality violation over inactive descendants in the lattice, add violating NCFs to U, and repeat until no inactive NCF violates the optimality condition.

C Proofs

C.1 Proof of Lemma 3.1

Proof. We use the following variational identity following from Hölder's inequality (Hardy et al., 1952): for any $a_1, \ldots, a_m \ge 0$ and $r > 0$,

$$\left(\sum_{i=1}^m a_i^{r/(r+1)}\right)^{(r+1)/r} = \min_{\theta\in\Theta_{m,r}} \sum_{i=1}^m \frac{a_i}{\theta_i}, \tag{C.1}$$

with equality attained when θ_i is proportional to $a_i^{1/(r+1)}$.

First, define the squared group magnitude $A_{u,g}(W) = \sum_{j \in g} \|W_{j,u}\|_2^2$. Then $\|\{W_{j,u}\}_{j \in g}\|_2 = A_{u,g}(W)^{1/2}$. Applying (C.1) with $r = 1$ to the geographical group term gives

$$\left(\sum_{g=1}^{G} \|\{W_{j,u}\}_{j \in g}\|_2 \right)^2 = \min_{\mu_{u,v} \in \Theta_{G,1}} \sum_{g=1}^{G} \frac{1}{\mu_{u,v,g}} \sum_{j \in g} \|W_{j,u}\|_2^2. \tag{C.2}$$

Next, define

$$B_u(W) = \sum_{g=1}^{G} \|\{W_{j,u}\}_{j \in g}\|_2 . \tag{C.3}$$

For each fixed v, the squared feature-hierarchy term in (2.6) is

$$\left(\sum_{u \in D(v)} B_u(W)^p \right)^{2/p}. \tag{C.4}$$

Since $1 < p \leq 2$, set $\bar{p} = p/(2-p)$. Applying (C.1) to (C.4) yields

$$\left(\sum_{u \in D(v)} B_u(W)^p \right)^{2/p} = \min_{\lambda_v \in \Theta_{|D(v)|,\bar{p}}} \sum_{u \in D(v)} \frac{B_u(W)^2}{\lambda_{v,u}}. \tag{C.5}$$

Substituting (C.2) into (C.5) gives

$$\left(\sum_{u \in D(v)} B_u(W)^p \right)^{2/p} = \min_{\lambda_v, \mu} \sum_{u \in D(v)} \frac{1}{\lambda_{v,u}} \sum_{g=1}^{G} \frac{1}{\mu_{u,v,g}} \sum_{j \in g} \|W_{j,u}\|_2^2. \tag{C.6}$$

Now define

$$C_v(W) = d_v \left(\sum_{u \in D(v)} B_u(W)^p \right)^{1/p}. \tag{C.7}$$

Then $\Omega(W) = \sum_{v \in \mathcal{V}} C_v(W)$. Applying (C.1) with $r = 1$ gives

$$\Omega(W)^2 = \left(\sum_{v \in \mathcal{V}} C_v(W) \right)^2 = \min_{\gamma \in \Theta_{|\mathcal{V}|,1}} \sum_{v \in \mathcal{V}} \frac{C_v(W)^2}{\gamma_v}. \tag{C.8}$$

By (C.7),

$$C_v(W)^2 = d_v^2 \left(\sum_{u \in D(v)} B_u(W)^p \right)^{2/p}. \tag{C.9}$$

Substituting (C.6) and (C.9) into (C.8) yields

$$\Omega(W)^2 = \min_{\gamma, \lambda, \mu} \sum_{v \in \mathcal{V}} \frac{d_v^2}{\gamma_v} \sum_{u \in D(v)} \frac{1}{\lambda_{v,u}} \sum_{g=1}^{G} \frac{1}{\mu_{u,v,g}} \sum_{j \in g} \|W_{j,u}\|_2^2, \tag{C.10}$$

which is exactly (3.2). For fixed γ, λ, μ, collect all terms in (3.2) that multiply the same block $\sum_{j \in g} \|W_{j,u}\|_2^2$. Define

$$\Psi_{u,g}(\gamma, \lambda, \mu) = \left(\sum_{v \in A(u)} \frac{d_v^2}{\gamma_v \lambda_{v,u} \mu_{u,v,g}} \right)^{-1}. \tag{C.11}$$

Then (3.2) can be rewritten as

$$\Omega(W)^2 = \min_{\gamma, \lambda, \mu} \sum_{u \in \mathcal{V}} \sum_{g=1}^{G} \Psi_{u,g}^{-1}(\gamma, \lambda, \mu) \sum_{j \in g} \|W_{j,u}\|_2^2. \tag{C.12}$$

Therefore, conditional on γ, λ, μ, the two-level group lasso regularizer in (2.6) reduces to a weighted ridge penalty over W. □

C.2 Proof of Proposition 3.2 (VQF ≡ MM)

We prove that the MM solver and the VQF solver generate identical W-iterates when they use the same exact variational weights. The proof follows from the fact that both methods reduce the HTNL regularizer to the same weighted-ridge subproblem after the auxiliary variables are fixed.

Proof. Define the variational quadratic form

$$R(W;\gamma,\lambda,\mu)=\sum_{v\in\mathcal{V}}\frac{d_v^2}{\gamma_v}\sum_{u\in D(v)}\frac{1}{\lambda_{v,u}}\sum_{g=1}^{G}\frac{1}{\mu_{u,v,g}}\sum_{j\in g}\|W_{j,u}\|_2^2. \tag{C.13}$$

By the variational identity in (3.2), the squared HTNL regularizer satisfies

$$\Omega(W)^2=\min_{\gamma,\lambda,\mu}R(W;\gamma,\lambda,\mu), \tag{C.14}$$

where $\gamma\in\Theta_{|\mathcal{V}|,1}$, $\lambda_v\in\Theta_{|D(v)|,\bar{p}}$, $\mu_{u,v}\in\Theta_{G,1}$, and $\bar{p}=p/(2-p)$.

At iteration t, both MM and VQF compute the same auxiliary variables by minimizing the variational form at the current iterate:

$$(\gamma^{(t+1)},\lambda^{(t+1)},\mu^{(t+1)})\in\arg\min_{\gamma,\lambda,\mu}R(W^{(t)};\gamma,\lambda,\mu). \tag{C.15}$$

For fixed $(\gamma^{(t+1)},\lambda^{(t+1)},\mu^{(t+1)})$, the regularizer becomes a quadratic penalty in W. Collecting all terms that multiply the same block $\sum_{j\in g}\|W_{j,u}\|_2^2$, define

$$\Psi_{u,g}^{(t+1)}=\left(\sum_{v\in A(u)}\frac{d_v^2}{\gamma_v^{(t+1)}\lambda_{v,u}^{(t+1)}\mu_{u,v,g}^{(t+1)}}\right)^{-1}. \tag{C.16}$$

Then the fixed-auxiliary regularizer is

$$R(W;\gamma^{(t+1)},\lambda^{(t+1)},\mu^{(t+1)})=\sum_{u\in\mathcal{V}}\sum_{g=1}^{G}\left(\Psi_{u,g}^{(t+1)}\right)^{-1}\sum_{j\in g}\|W_{j,u}\|_2^2. \tag{C.17}$$

Thus, the W-update solved by MM is

$$W_{\mathrm{MM}}^{(t+1)}=\arg\min_{W}\mathcal{L}(W)+\lambda_{\mathrm{reg}}\sum_{u\in\mathcal{V}}\sum_{g=1}^{G}\left(\Psi_{u,g}^{(t+1)}\right)^{-1}\sum_{j\in g}\|W_{j,u}\|_2^2. \tag{C.18}$$

The VQF solver uses the same quadratic representation in (C.14). With the same auxiliary variables from (C.15), its W-update is

$$W_{\mathrm{VQF}}^{(t+1)}=\arg\min_{W}\mathcal{L}(W)+\lambda_{\mathrm{reg}}\sum_{u\in\mathcal{V}}\sum_{g=1}^{G}\left(\Psi_{u,g}^{(t+1)}\right)^{-1}\sum_{j\in g}\|W_{j,u}\|_2^2. \tag{C.19}$$

The optimization problems in (C.18) and (C.19) are identical. Therefore,

$$W_{\mathrm{MM}}^{(t+1)}=W_{\mathrm{VQF}}^{(t+1)}. \tag{C.20}$$

Since both methods start from the same initialization $W^{(0)}$, the same auxiliary variables are computed at the first iteration, and the same W-update is solved. By induction, if $W_{\mathrm{MM}}^{(t)}=W_{\mathrm{VQF}}^{(t)}$, then (C.15) gives the same auxiliary variables and (C.20) gives $W_{\mathrm{MM}}^{(t+1)}=W_{\mathrm{VQF}}^{(t+1)}$. Hence the two methods generate identical iterates for all $t\geq 0$. □

The proposition proves mathematical equivalence under exact auxiliary-variable updates and the same numerical convention for zero-norm blocks. In our implementation, MM and VQF call the same weighted-ridge solver after constructing the same block weights $\Psi_{u,g}^{(t+1)}$. This explains the observed bit-exact agreement of their selected NCF sets in the real-data experiments.

C.3 Proof of Proposition 3.4 (Inexact ADMM Residual)

Proof. Let

$$\mathcal{F}_\rho(Z) = \lambda\Omega^2(Z) + \frac{\rho}{2}\|Z - Q\|_2^2. \tag{C.21}$$

The regularizer Ω^2 is convex, and the quadratic term is ρ-strongly convex. Therefore, $\mathcal{F}_\rho$ is also ρ-strongly convex. By strong convexity, for any Z and the unique minimizer Z^*,

$$\mathcal{F}_\rho(Z) - \mathcal{F}_\rho(Z^*) \geq \frac{\rho}{2}\|Z - Z^*\|_2^2. \tag{C.22}$$

Substituting $Z = Z^{(K)}$ gives

$$\varepsilon_K = \mathcal{F}_\rho(Z^{(K)}) - \mathcal{F}_\rho(Z^*) \geq \frac{\rho}{2}\|Z^{(K)} - Z^*\|_2^2. \tag{C.23}$$

Rearranging yields

$$\|Z^{(K)} - Z^*\|_2 \leq \sqrt{\frac{2\varepsilon_K}{\rho}}. \tag{C.24}$$

If the inner solver satisfies the standard accelerated objective rate $\varepsilon_K = O(K^{-2})$, then

$$\|Z^{(K)} - Z^*\|_2 = O\left(\frac{1}{\sqrt{\rho}K}\right). \tag{C.25}$$

□

C.4 Proof of Zero-Block Recovery in ADMM

Proof. For the proximal problem in (3.7), first-order optimality gives

$$0 \in \rho(Z^* - Q) + \lambda\partial\Omega^2(Z^*). \tag{C.26}$$

Restricting (C.26) to block u and considering $Z_u^* = 0$ yields

$$\rho Q_u \in \lambda\partial_u\Omega^2(0; Z_{-u}^*). \tag{C.27}$$

Thus, exact zero-block recovery is determined by whether ρQ_u lies inside the corresponding block-wise subgradient set. Define the distance to its zero/nonzero boundary as

$$m_u := \operatorname{dist}\big(\rho Q_u, \partial\big[\lambda\partial_u\Omega^2(0; Z_{-u}^*)\big]\big). \tag{C.28}$$

If the perturbation induced by the approximate proximal solution is smaller than m_u, it cannot cross this boundary, so $Z^{(K)}$ and Z^* have the same zero/nonzero status for block u up to numerical tolerance. By Proposition 3.4, a sufficient condition is therefore

$$\sqrt{\frac{2\varepsilon_K}{\rho}} < m_u. \tag{C.29}$$

For the hierarchical penalty, a useful local scale is

$$\rho_c^{\text{proxy}} = \frac{d_v}{\|f_v(W)^{p-1}\|_\infty}, \qquad f_v(W) = \left(\sum_{u \in D(v)} \|W_u\|_2^p\right)^{1/p}. \tag{C.30}$$

where $d_v = 2^{|v|}$ and $p \in (1, 2]$. This is a local heuristic scale rather than a universal critical value; the exact zero-block decision remains governed by (C.27). Hence, once the finite-step residual falls below the local KKT margin, the approximate and exact proximal updates recover the same support up to numerical tolerance. □

C.5 Proof of Proposition 3.7 (Active-Set KKT Ceiling)

Proof. For each pair (s,t), define

$$g_{st} := \nabla_W \mathcal{L}(W^*)_{st}. \tag{C.31}$$

Using the triangle inequality and the gradient bound in (3.17), we have

$$\mathrm{score}(v) = \frac{1}{d_v}\left\|\sum_{s,t} g_{st}\phi_v(X_{st})\right\|_2 \le \frac{1}{d_v}\sum_{s,t}\|g_{st}\|_2|\phi_v(X_{st})| \le \frac{G}{d_v}\sum_{s,t}\phi_v(X_{st}). \tag{C.32}$$

Since $\phi_v(X_{st})$ is a product of independent Bernoulli variables,

$$\mathbb{E}[\phi_v(X_{st})] = \mathbb{E}\left[\prod_{j\in v} x_j(s,t)\right] = \prod_{j\in v}\mathbb{E}[x_j(s,t)] = \prod_{j\in v} p_j \le p_{\max}^k. \tag{C.33}$$

Taking expectation in (C.32) gives

$$\mathbb{E}[\mathrm{score}(v)] \le \frac{G}{d_v}\sum_{s,t}\mathbb{E}[\phi_v(X_{st})] \le \frac{Gn_{\mathrm{eff}}p_{\max}^k}{2^k} = Gn_{\mathrm{eff}}\left(\frac{p_{\max}}{2}\right)^k. \tag{C.34}$$

This proves (3.18).

To obtain the threshold length, require

$$Gn_{\mathrm{eff}}\left(\frac{p_{\max}}{2}\right)^k \le \tau_{\mathrm{KKT}}. \tag{C.35}$$

Since $0 < p_{\max}/2 < 1$, this condition is equivalent to

$$k \ge \frac{\log(Gn_{\mathrm{eff}}/\tau_{\mathrm{KKT}})}{\log(2/p_{\max})}. \tag{C.36}$$

Taking the ceiling gives the k^* condition.

Finally, since $\mathrm{score}(v) \ge 0$, Markov's inequality gives

$$\Pr\left[\mathrm{score}(v) \ge \tau_{\mathrm{KKT}} + \epsilon\right] \le \frac{\mathbb{E}[\mathrm{score}(v)]}{\tau_{\mathrm{KKT}}+\epsilon} \le \frac{Gn_{\mathrm{eff}}\left(p_{\max}/2\right)^k}{\tau_{\mathrm{KKT}}+\epsilon}. \tag{C.37}$$

This proves the exponential decay claim. □

C.6 Proof of Corollary 3.8

Proof. The active-set rule can only add NCFs that are considered by the search and pass the KKT threshold. Any $u \in U^*$ with $|u| > D$ is excluded by the maximum-length constraint. Any $u \in U^*$ with $\mathbb{E}[\mathrm{score}(u)] \le \tau_{\mathrm{KKT}}$ is below the expected expansion threshold by Proposition 3.7. Therefore, in expectation, the recoverable true support is contained in U^*_{acc}. Since

$$\mathrm{Recall}(\hat{S}; U^*) = \frac{|\hat{S}\cap U^*|}{|U^*|}, \tag{C.38}$$

we obtain

$$\mathbb{E}\left[\mathrm{Recall}(\hat{S}; U^*)\right] = \frac{\mathbb{E}[|\hat{S}\cap U^*|]}{|U^*|} \le \frac{|U^*_{\mathrm{acc}}|}{|U^*|}. \tag{C.39}$$

□

D Additional Diagnostics

D.1 WGL Regularization-Scale Sensitivity

We investigate the sensitivity of WGL to the regularization scale. Under the default continuation schedule ($\lambda : 500 \to 50$), WGL collapses to an empty selected set with AUC $= 0.500$ on all 10 civil-unrest datasets, whose positive-class fractions range from 0.5% to 1.5%. In contrast, the flu dataset,

with a positive-class fraction of 7.4%, does not fully collapse. This pattern suggests that the default WGL scale is particularly aggressive in the severely imbalanced setting, although the comparison does not establish class imbalance as the sole cause. The relationship between positive-class fraction and WGL behavior is further examined in Appendix F.5.

To determine whether the failure reflects the WGL formulation itself or its regularization scale, we vary λ while keeping the remaining training configuration fixed. Table 3 reports a representative sweep on Venezuela (positive rate 0.7%). The default $500 \to 50$ continuation produces an empty support and chance-level AUC. Reducing the regularization scale restores nonzero supports and predictive performance: a fixed $\lambda = 1$ selects seven NCFs and increases AUC from 0.500 to 0.581. Further decreasing λ to 0.1 increases the selected set to 13 NCFs but reduces AUC to 0.536, showing that weaker regularization does not necessarily improve predictive performance.

These results identify regularization scale as an important factor in WGL behavior. The default collapse therefore should not be interpreted as an inherent inability of WGL to select predictive hierarchical features. Based on this diagnostic, we use WGL($\lambda = 1$) as the non-collapsing WGL configuration in the main predictive comparison.

Table 3: Sensitivity of WGL to the regularization scale on Venezuela. "Direction" indicates either high-to-low continuation from λ_{start} to λ_{end} or a single fixed-λ run. "sel@0.40" denotes the number of NCFs selected at the active-set threshold $\tau = 0.40$.

λ_{start}	λ_{end}	Direction	AUC	sel@0.40
500	50	high-to-low	0.500	0
50	5	high-to-low	0.572	1
5	0.5	high-to-low	0.540	9
1	**1**	single	**0.581**	**7**
0.1	0.1	single	0.536	13

D.2 Paired-Seed Analysis of the ADMM Proximal Update

To examine whether the aggregate improvement from bisection is consistent across runs, Figure 4 compares ADMM-Bisect and ADMM-FISTA seed by seed at $\rho = 1$. ADMM-Bisect matches or improves ADMM-FISTA on every paired seed in this diagnostic, with most points lying above the identity diagonal. Together with the Holm-survived aggregate improvement ($\Delta\text{F1} = +0.088$, $p_{\text{adj}} = 2.60 \times 10^{-3}$), this result provides consistent empirical support for replacing the finite-step FISTA proximal update with bisection.

E Active-Set Mechanism Analysis

All HTNL optimization methods studied in this paper share the same outer active-set procedure. At each iteration, inactive candidate NCFs are scored using the KKT criterion in Equation 2.9, the top-K candidates are added to the active set, and the restricted optimization problem is re-solved. The final selected set $\hat{S}$ is determined by the threshold τ. Because this outer procedure is shared across methods, differences in selected NCFs arise from the inner optimization and its effect on subsequent KKT scores.

E.1 Per-Method Active-Set Behavior

The optimization methods exhibit distinct selection behavior. **MM/VQF** produces compact and stable supports: the two methods select identical NCF sets on all real-data benchmarks, with Jaccard similarity 1.000, consistent with Proposition 3.2. At $\tau = 0.40$, they typically select about 4–5 NCFs on the LatAm datasets. **ADMM-Bisect** selects broader supports. Although bisection removes the finite-step FISTA proximal error in the ADMM Z-update, its selected sets remain larger and have low overlap with MM/VQF, showing that a more accurate proximal update does not recover the same compact support. **WGL** is sensitive to the regularization scale: the default continuation path $\lambda_{\text{init}} = 500 \to \lambda_{\text{target}} = 50$ produces empty supports on the LatAm datasets, whereas $\lambda = 1$ restores nonzero supports. Under this setting, WGL is closer to ADMM-Bisect than to MM/VQF,

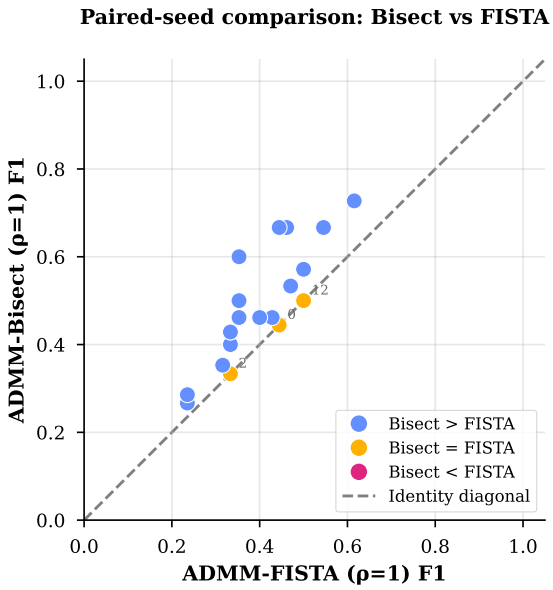


Figure 4: Paired-seed F1 comparison between ADMM-Bisect and ADMM-FISTA at $\rho = 1$. Each point represents the same random seed evaluated with the two proximal solvers; points above the identity diagonal favor ADMM-Bisect. Under this diagnostic condition, ADMM-Bisect matches or improves ADMM-FISTA on every evaluated seed, showing that the aggregate improvement is consistent across paired runs rather than driven by a small number of seeds.

with Jaccard similarities of 0.780 and 0.351, respectively. **WGL-Adaptive** instead produces very small selected sets, indicating that its adaptive weighting can strongly suppress weaker candidates.

E.2 Comparative Summary

Table 4 summarizes the selected-support behavior on the LatAm datasets. MM/VQF produces the most compact supports, selecting approximately 4–5 NCFs while achieving the largest mean AUC improvement. ADMM-Bisect and WGL($\lambda = 1$) select broader supports of approximately 10 NCFs and show substantially lower overlap with MM/VQF. In contrast, the default WGL continuation produces empty supports. These results show that different inner optimization methods can lead to substantially different selected sets even when they share the same outer active-set procedure.

Table 4: Selected-support behavior on the LatAm civil-unrest datasets. $|\hat{S}|$ denotes the selected set size at $\tau = 0.40$, Jaccard similarity is measured against MM/VQF, and ΔAUC is measured against class-weighted Logistic Regression.

Method	**Jac. vs Orig./VQF**	$\lvert\hat{S}\rvert$	**ΔAUC**
MM/VQF	1.000	≈4–5	$+0.074^{***}$
ADMM-Bisect	0.277	≈10	$+0.034^{*}$
WGL($\lambda = 1$)	0.351	≈10	$+0.044^{**}$
WGL-Cont (default)	0.000	0	—

E.3 Active-Set Depth Ceiling

The shared active-set rule imposes a structural limitation. Proposition 3.7 shows that the KKT score for a length-k NCF decays as $O((p_{\max}/2)^k)$, making sufficiently long NCFs increasingly unlikely to enter the active set for fixed n_{eff} and $\tau_{\text{KKT}} > 0$. Thus, improving the inner optimization method can change support size, predictive performance, and runtime, but does not directly remove the KKT-based candidate-admission limit.

Table 5 provides empirical evidence for this limitation. Across expanded active-set configurations, VQF remains at an F1 plateau of 0.686, while ADMM-Bisect varies only from 0.674 to 0.685. Increasing the active-set budget therefore does not overcome the observed ceiling, consistent with Proposition 3.7. Overcoming this limitation requires modifying the candidate-expansion mechanism rather than only the restricted inner optimization.

Table 5: Active-set sensitivity on the TKDE-faithful synthetic generator (Dataset 2). m_0 denotes the initial active-set size; the default configuration uses $m_0 = 2$ and adds five candidates per iteration. Entries report mean F1.

Method	**Default ($m_0 = 2$, add=5)**	$m_0 = 3$	$m_0 = 4$	$m_0 = 5$
MM/VQF	0.590	0.686	0.686	0.686
ADMM-Bisect	0.674	0.684	0.685	0.685

E.4 Selected-Support Differences Across Optimization Methods

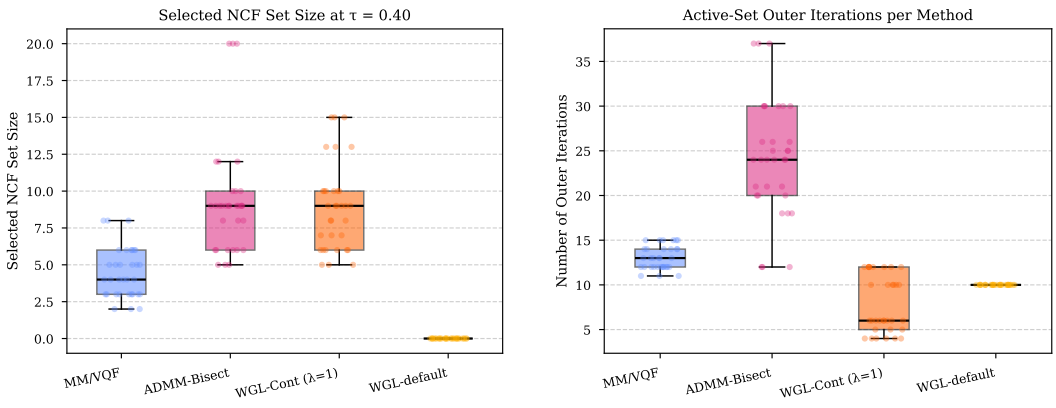
We compare the NCF sets selected by different optimization methods. Table 6 reports the mean Jaccard similarity at $\tau = 0.40$ across the 11 real-data benchmarks.

Table 6: Mean Jaccard similarity between selected NCF sets across the 11 real-data benchmarks at $\tau = 0.40$. MM and VQF have Jaccard similarity 1.000, consistent with Proposition 3.2.

	Orig	**VQF**	**ADMM**	**WGL-def.**	**WGL($\lambda = 1$)**
MM	**1.000**	**1.000**	0.277	0.045	0.351
VQF	**1.000**	**1.000**	0.277	0.045	0.351
ADMM-Bisect	0.277	0.277	**1.000**	0.000	0.780
WGL-Cont (def.)	0.045	0.045	0.000	**1.000**	0.000
WGL($\lambda = 1$)	0.351	0.351	0.780	0.000	**1.000**

MM and VQF select identical supports, as predicted by Proposition 3.2. ADMM-Bisect has substantially lower agreement with MM/VQF (0.277), showing that correcting the proximal update does not make ADMM select the same compact support. After regularization-scale correction, WGL($\lambda = 1$) is more similar to ADMM-Bisect than to MM/VQF, with Jaccard similarities of 0.780 and 0.351, respectively. Thus, ADMM-Bisect and calibrated WGL tend to select similar, broader NCF sets.

E.5 Selection Size and Outer Iterations



(a) Selected NCF set size at $\tau = 0.40$.

(b) Number of outer active-set iterations.

Figure 5: Selection size and outer active-set iterations across optimization methods. Each point represents one dataset–seed pair. MM/VQF produces more compact selected sets, whereas ADMM-Bisect and WGL($\lambda = 1$) retain broader supports. ADMM-Bisect requires more outer iterations but remains computationally efficient in the tested implementation.

Figure 5 compares selected-set size and the number of outer active-set iterations. MM/VQF produces the most compact supports, with a median of four selected NCFs and a standard deviation of 1.7. ADMM-Bisect and WGL($\lambda = 1$) both have a median selected-set size of nine, with standard deviations of 3.9 and 2.9, respectively. Thus, the broader supports of ADMM-Bisect and calibrated WGL are observed consistently across datasets and seeds. Default WGL instead produces empty supports on the 10 LatAm benchmarks.

The number of outer iterations shows a different pattern. MM/VQF requires approximately 13 iterations on average, compared with approximately 24 for ADMM-Bisect and 10 for WGL($\lambda = 1$). Despite requiring more outer iterations, ADMM-Bisect remains faster in wall-clock time because its tested implementation reduces the cost of each iteration through KKT pruning. This distinction shows that the number of outer iterations alone does not determine total runtime.

F Additional Real-Data Results

F.1 VQF ≡ MM on Real Data

Proposition 3.2 shows that VQF and MM perform the same weighted-ridge updates under matched auxiliary updates. This equivalence is also observed empirically on all 11 real-data benchmarks: VQF and MM select identical NCF sets, with Jaccard similarity 1.000 across all evaluated runs. We therefore report the two methods jointly as MM/VQF in the remaining real-data analyses.

F.2 Real-Data Selection and Predictive Performance

Although MM/VQF produces identical selected sets, ADMM-Bisect exhibits substantially different selection behavior. As shown in Table 7, ADMM-Bisect selects 62–97 NCFs on the five TKDE-replication civil-unrest datasets, compared with 13–25 for MM/VQF. The mean selected-support size is 80.8 for ADMM-Bisect and 18.6 for MM/VQF. Thus, replacing the finite-step FISTA proximal update with bisection does not make the ADMM solution recover the more compact MM/VQF support. On the same five datasets, ADMM-Bisect also requires less runtime, averaging 5.6 seconds per country compared with 25.5 seconds for MM/VQF.

Figure 6 summarizes predictive performance across all 11 datasets. MM/VQF improves over class-weighted Logistic Regression on every dataset, while ADMM-Bisect and WGL($\lambda = 1$) also yield positive mean AUC gains. The broader supports selected by ADMM-Bisect and WGL($\lambda = 1$) do not translate into higher AUC than the more compact MM/VQF solutions. In contrast, the default WGL continuation schedule produces empty models on the 10 severely imbalanced civil-unrest datasets, showing its sensitivity to the regularization scale. We use AUC as the primary predictive metric because these datasets are highly imbalanced.

Table 7: Selected-support sizes on five civil-unrest datasets. n_{train} and pos% denote the number of training instances and the positive-class fraction, respectively. Support size is measured at $\tau = 0.40$. MM and VQF are identical under matched updates and are therefore reported jointly. ADMM-Bisect uses $\rho = 1$.

Country	n_{train}	**pos%**	**MM/VQF**	**ADMM-Bisect**
Uruguay	10,788	0.8%	13	76
Venezuela	11,832	0.7%	15	62
Paraguay	14,964	1.5%	22	86
Colombia	19,140	0.5%	25	97
Argentina	19,836	0.5%	18	83
Mean	—	—	**18.6**	**80.8**

F.3 Runtime and Efficiency Analysis

Figure 7a reports the mean wall-clock time of each optimization method on the 10 LatAm civil-unrest datasets and the flu dataset. All runs use a single CPU process on the same machine.

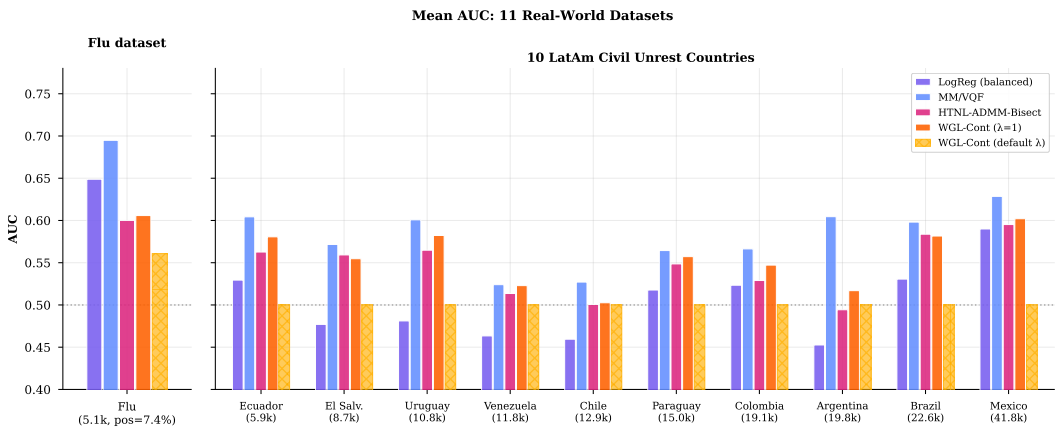


Figure 6: Predictive AUC across the 11 real-data benchmarks. MM/VQF provides the largest mean improvement over class-weighted Logistic Regression. ADMM-Bisect and WGL($\lambda = 1$) also improve mean AUC but retain broader selected supports, while the default WGL continuation produces empty models on the severely imbalanced civil-unrest datasets.

On the 10 LatAm datasets, MM and VQF have nearly identical runtimes, averaging approximately 25 s per dataset. ADMM-Bisect averages approximately 6 s, corresponding to a 3–4$\times$ speedup. The tested implementation combines bisection with KKT pruning, which removes candidates with scores below $\tau_{\text{prune}} = 0.10$ before the inner ADMM solve and reduces the size of the resulting subproblem (Section 3.5 and Appendix 5.3).

WGL-Cont has greater runtime variation. It is fast when the continuation procedure terminates early with an empty support, but becomes slower when multiple λ stages are evaluated. On Mexico, the largest LatAm dataset ($n = 41{,}760$), WGL-Cont takes 140 s, compared with approximately 25 s for MM.

On the flu dataset (pos$\% = 7.4\%$), ADMM-Bisect takes approximately 18 s and MM approximately 22 s. WGL($\lambda = 1$) is fastest at approximately 9 s because the fixed-λ configuration avoids the multi-stage continuation procedure.

Overall, ADMM-Bisect remains computationally efficient despite using an accurate proximal update. The scalar bisection solver requires only $O(\log(1/\varepsilon))$ iterations, while KKT pruning further reduces the number of blocks processed by the inner solver.

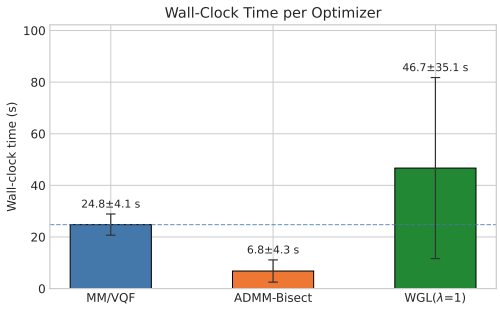
F.4 Per-Dataset AUC

Figure 7b shows the per-dataset AUC results. MM/VQF achieves higher AUC than ADMM-Bisect and WGL($\lambda = 1$) on all 10 LatAm datasets, with particularly large improvements over Logistic Regression on Argentina ($+0.152$) and Uruguay ($+0.120$). ADMM-Bisect and WGL($\lambda = 1$) generally remain closer to the Logistic Regression baseline and also select broader NCF sets. Thus, the additional features retained by these methods do not produce higher AUC on the LatAm benchmarks.

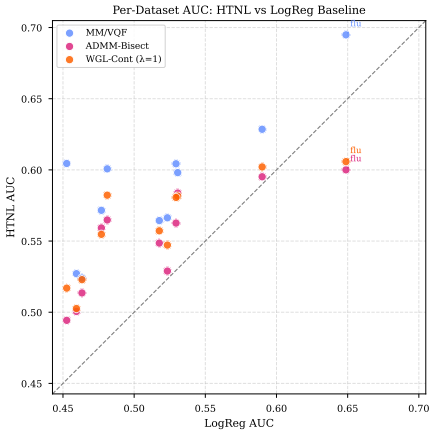
The flu dataset differs from the LatAm datasets in both positive-class fraction and performance pattern. Its positive rate is 7.4%, compared with 0.48–1.52% for the LatAm datasets, and the differences among the HTNL methods are smaller. Because only one dataset lies outside the severely imbalanced LatAm range, we treat this pattern as descriptive rather than evidence for a specific class-imbalance threshold.

F.5 Performance across Class-Imbalance Levels

Figure 8 examines performance as a function of the positive-class fraction. On all 10 severely imbalanced LatAm datasets, default WGL produces empty supports and AUC $= 0.500$, whereas MM/VQF, ADMM-Bisect, and WGL($\lambda = 1$) produce nonzero solutions. MM/VQF achieves the highest AUC among the evaluated HTNL methods on each of these datasets. This pattern shows



(a) Mean wall-clock time. Error bars show ± 1 standard deviation across datasets.



(b) Per-dataset AUC relative to Logistic Regression. Points above the diagonal indicate higher AUC than the baseline.

Figure 7: Runtime and predictive performance across the 10 LatAm civil-unrest datasets and the flu dataset. The left panel compares wall-clock time, and the right panel compares per-dataset AUC with class-weighted Logistic Regression.

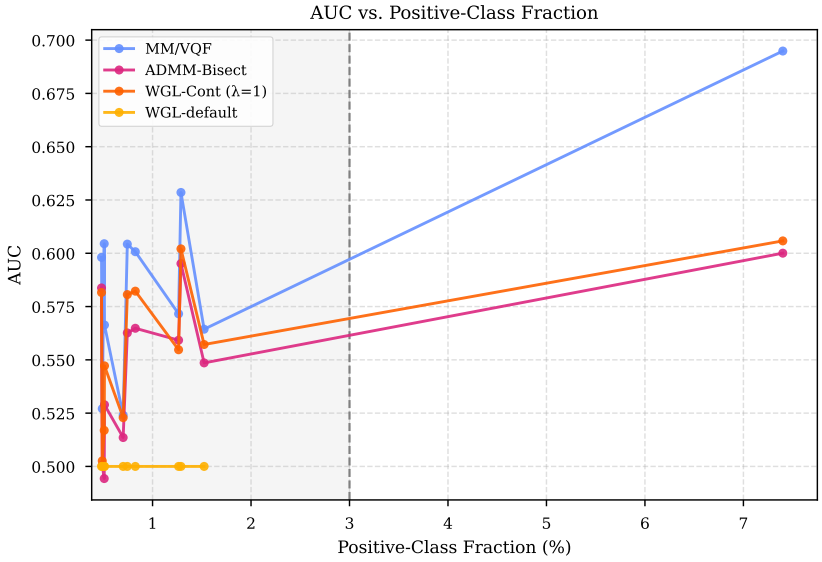


Figure 8: AUC as a function of the positive-class fraction. The 10 LatAm datasets have positive rates between 0.48% and 1.52%, whereas the flu dataset has a positive rate of 7.4%. Default WGL produces empty supports with AUC $= 0.500$ on all 10 LatAm datasets, while this failure is not observed on the flu dataset.

that the default WGL regularization scale is poorly matched to the LatAm setting, while reducing λ restores nonzero solutions.

The flu dataset, with a positive rate of 7.4%, does not show the same default-WGL failure pattern, and the AUC differences among the non-collapsing methods are smaller. This contrast suggests an association between the data scale, class imbalance, and WGL sensitivity, but the current benchmarks do not identify a specific positive-rate boundary or establish class imbalance as the sole cause of the collapse.

G SYSTEMATIC OPTIMIZATION METHOD COMPARISON

The four HTNL optimization families studied in this paper differ along several technical dimensions, including objective formulation, inner optimization routine, convergence behavior, per-iteration computational cost, active-set selection behavior, robustness to class imbalance, and hyperparameter requirements. This section systematically compares these differences, connects them to the mechanisms identified in our analysis, and summarizes practical guidelines for choosing an optimization method.

G.1 MULTI-DIMENSIONAL OVERVIEW

Table 8 compares the four optimization families across eight dimensions. MM and VQF use the same variational quadratic formulation and produce identical updates, whereas ADMM-Bisect directly handles the nonsmooth objective through ADMM and WGL converts the hierarchical penalty into a sequence of weighted group-lasso problems. These different optimization strategies lead to distinct convergence, computational, and feature-selection behavior.

Table 8: Multi-dimensional comparison of the four HTNL optimization families. "Sparse" denotes the sparse-selection regime ($\approx$4–5 selected NCFs), "broad" denotes the broader-selection regime ($\approx$10 selected NCFs), and "collapse" denotes an empty or nearly empty selected set. "Ridge" refers to the weighted ridge subproblem obtained after fixing the variational auxiliary variables, and "FISTA" refers to the accelerated proximal-gradient solver used for the weighted group-lasso subproblem. The parameter $\rho \in \mathbb{R}_+$ is the ADMM augmented-Lagrangian penalty, while the λ schedule denotes the continuation path used by WGL.

Dimension	MM	VQF	ADMM-Bisect	WGL-Cont
Formulation basis	Variational quadratic formulation	Same variational quadratic formulation as MM	ADMM decomposition of the original nonsmooth objective	Reformulation of the outer ℓ_p hierarchy as weighted group lasso
Inner solver	Weighted ridge subproblem	Same weighted ridge subproblem as MM	L-BFGS for the smooth step plus exact bisection for the nonsmooth proximal step	FISTA for the weighted group-lasso subproblem
Convergence behavior	Linear convergence for the strongly convex ridge subproblem	Same as MM	$O(1/K)$ ADMM residual convergence	$O(1/K^2)$ convergence within each fixed-λ FISTA stage; no recovery guarantee after support collapse
Per-iteration cost	$O(\lvert U\rvert \cdot \text{ridge})$	$O(\lvert U\rvert \cdot \text{ridge})$	$O(\lvert U\rvert \cdot \lvert D(v)\rvert \cdot \log(1/\varepsilon))$	$O(\lvert U\rvert \cdot G \cdot \text{FISTA})$
Selection regime	Sparse: $\approx$4–5 NCFs	Identical to MM	Broad: $\approx$10 NCFs	Broad when λ=1; collapse under the default continuation schedule
Class-imbalance robustness	Robust in the tested rare-event datasets	Robust in the tested rare-event datasets	Robust in the tested rare-event datasets	Scale-sensitive; the default schedule collapses on all 10 severely imbalanced civil-unrest datasets
Extra hyperparameters	None beyond shared C and τ	None beyond shared C and τ	ρ, the augmented-Lagrangian penalty	λ_{init}, λ_{target}, and the number of continuation steps
When to use	Sparse, interpretable selection with minimal tuning	Same as MM, since the updates are algebraically equivalent	Broader support recovery with exact proximal computation and stable AUC across tested ρ values	Broader support or regularization-path analysis; requires λ calibration on severely imbalanced data

G.2 FORMULATION EQUIVALENCES AND DISTINCTIONS

The four methods can be grouped into three formulations. MM and VQF use the same variational quadratic formulation; ADMM-Bisect directly optimizes the original nonsmooth objective using ADMM; and WGL rewrites the hierarchical penalty as a weighted one-level group-lasso problem. These differences lead to distinct optimization behavior, failure modes, and selected feature sets.

MM/VQF: equivalent weighted-ridge updates. MM and VQF introduce auxiliary variables (γ, λ, μ) to express the two-level HTNL regularizer as a weighted quadratic objective; see Lemma 3.1. Under matched initialization and exact auxiliary updates, Proposition 3.2 shows that they solve the same weighted-ridge subproblem at every iteration. We therefore treat them as one method family, VQF. The weighted-ridge update strongly shrinks marginally active weights toward zero, leading VQF to select compact supports in our experiments. This behavior makes it suitable when sparse, interpretable feature selection and minimal tuning are preferred over recovering a broader set of features.

ADMM-Bisect: direct nonsmooth optimization with exact proximal updates. ADMM-Bisect retains the original two-level group-norm regularizer and uses ADMM to separate the smooth loss from the nonsmooth regularization; see Equation B.6. The W-update optimizes the smooth loss with a quadratic agreement penalty, while the Z-update applies the proximal operator of the local two-level regularizer. ADMM-Bisect replaces the truncated FISTA approximation in the Z-update with a bisection solver. Proposition 3.4 shows that finite-step proximal error can leave blocks near the zero/nonzero boundary incorrectly nonzero, whereas bisection solves the proximal equations to numerical tolerance. ADMM-Bisect nevertheless selects broader supports than VQF because its intermediate supports can increase the KKT scores of related inactive candidates. It is therefore useful when accurate proximal computation and broader feature recovery are desired.

WGL: weighted group-lasso reformulation. WGL uses the dual representation of the outer ℓ_p norm to rewrite the two-level hierarchy as a weighted one-level group-lasso problem; see Equation B.21. For fixed weights, the resulting group-lasso problem can be solved efficiently, after which the weights are updated to reflect the hierarchical penalty. The main practical limitation is sensitivity to the regularization scale. When the initial λ is too large, the first continuation stage can shrink all groups to zero, making subsequent recovery difficult. In our experiments, the default schedule $\lambda_{\text{init}} = 500 \to \lambda_{\text{target}} = 50$ collapses on all 10 severely imbalanced civil-unrest datasets, whereas $\lambda = 1$ restores nonzero supports and predictive performance. WGL therefore requires careful λ calibration on severely imbalanced data.

G.3 Convergence and Computational Cost

The optimization families differ substantially in both convergence behavior and wall-clock cost. Faster convergence of an individual inner problem does not necessarily lead to lower total runtime, because the overall cost also depends on the number of active blocks, candidate screening, and repeated active-set updates.

VQF: fast inner convergence but moderate overall cost. Each outer iteration of VQF solves a strongly convex weighted ridge subproblem, for which standard first-order solvers achieve linear convergence under the usual smoothness and strong-convexity conditions. The per-iteration cost is $O(|U| \cdot P^2)$. Across the 11 benchmarks, VQF takes $26.7\,\text{s}$ on average, making it the slowest non-collapsing variant in our comparison. Much of this cost comes from repeatedly solving the ridge problem over the full active set, including blocks whose weights remain small.

ADMM-Bisect: slower theoretical rate but lower total runtime. For fixed ρ, standard ADMM has an $O(1/K)$ convergence rate under appropriate convexity assumptions (Boyd et al., 2011). Despite this slower theoretical rate, ADMM-Bisect requires only $19.8\,\text{s}$ on average, compared with $26.7\,\text{s}$ for VQF, corresponding to a $1.3\times$ overall speedup on the same hardware. We further evaluate a computational variant that combines KKT pruning with the bisection proximal solver. On the synthetic diagnostic, this configuration preserves essentially the same F1 as baseline ADMM while reducing wall-clock time from approximately $15\,\text{s}$ to $4\,\text{s}$, corresponding to a $3.9\times$ speedup (Appendix 5.3). Thus, practical efficiency depends not only on the convergence rate of the inner solver but also on reducing unnecessary computation during optimization.

WGL: fast inner optimization but sensitive continuation. For fixed λ, the FISTA solver used by WGL achieves an $O(1/K^2)$ objective convergence rate under standard assumptions (Beck & Teboulle, 2009). This guarantee applies to each fixed-λ subproblem rather than to the complete continuation procedure. In our experiments, a large initial λ can produce an empty support on the severely imbalanced civil-unrest datasets, whereas a calibrated regularization scale restores non-

degenerate solutions. WGL therefore combines efficient fixed-λ optimization with greater sensitivity to the regularization schedule and requires explicit λ calibration in the severely imbalanced setting.

G.4 Selection Behavior and Support Recovery

The optimization affects not only predictive performance but also the size and composition of the selected NCF set. Among the non-collapsing methods, VQF consistently selects compact supports, whereas ADMM-Bisect and WGL(λ=1) retain substantially broader supports. The default WGL configuration represents a separate failure mode in which the selected support becomes empty.

Compact selection: VQF. VQF produces the smallest selected sets, with approximately 4–5 NCFs per dataset at τ=0.40. This compact selection does not reduce predictive performance: VQF achieves the largest improvement over Logistic Regression (ΔAUC=+0.074, p=4×10^{-5}). MM and VQF also select identical supports on all 11 datasets, with Jaccard similarity 1.000, consistent with their equivalent weighted-ridge updates. VQF is therefore well suited when a small and interpretable set of selected indicators is desired.

Broader selection: ADMM-Bisect and WGL(λ=1). ADMM-Bisect and WGL(λ=1) select approximately 10 NCFs on average, roughly twice the support size of VQF. Their selected sets are also similar to each other, with Jaccard similarity 0.780, the largest off-diagonal value in the agreement matrix. Thus, after correcting the WGL regularization scale, both methods exhibit a similar broad-selection pattern.

For ADMM-Bisect, the broader support cannot be explained solely by an inaccurate proximal update. Replacing truncated FISTA with bisection removes the proximal residual described in Proposition 3.4, but the selected support remains broader than that of VQF. A contributing mechanism is the iterative active-set update: retaining more features in one round changes the fitted model and can increase the KKT scores of related inactive candidates, allowing additional candidates to enter in subsequent rounds. ADMM-Bisect and WGL(λ=1) are therefore useful when broader feature coverage is preferred to highly compact selection.

Support collapse: default WGL. Under the default continuation schedule, WGL returns an empty support with AUC = 0.500 on all 10 civil-unrest datasets, whose positive-class fractions are at most 1.52%. In contrast, the flu dataset, with a positive-class fraction of 7.4%, does not fully collapse. Together with the λ sensitivity analysis in Appendix D.1, these results indicate that WGL is particularly sensitive to the regularization scale in the severely imbalanced setting. WGL-Adaptive exhibits a related failure pattern, producing very small supports as its adaptive weighting increasingly favors already dominant candidates over weaker ones. These results motivate explicit λ calibration before applying WGL to severely imbalanced datasets.

G.5 Hyperparameter Sensitivity and Robustness

The methods differ substantially in their tuning requirements. VQF introduces no optimization-specific hyperparameters, ADMM-Bisect adds the ADMM penalty ρ, and WGL requires careful calibration of the regularization scale and continuation schedule.

VQF: no optimization-specific tuning. VQF uses only the shared penalty C and active-set threshold τ, which are also required by the other methods and can be selected using validation AUC. It introduces no additional optimization-specific hyperparameters, making it particularly suitable when the available tuning budget is limited.

ADMM-Bisect: limited sensitivity to ρ. ADMM-Bisect additionally requires the augmented-Lagrangian penalty ρ. In our sensitivity study, varying ρ over $\{0.1, 0.5, 1.0, 2.0\}$ changes AUC by less than 0.010 on the LatAm benchmarks. This indicates that, within the tested range, ρ primarily controls optimization behavior while predictive performance remains stable. We therefore use ρ=1.0 as the default across all 11 benchmarks. ADMM-Bisect provides broader support recovery at the cost of one additional tuning parameter.

WGL: strong sensitivity to λ. WGL is considerably more sensitive to its regularization scale. The default continuation schedule $\lambda_{\text{init}}=500 \rightarrow \lambda_{\text{target}}=50$ produces empty supports on all 10 severely imbalanced civil-unrest datasets, whereas reducing the regularization scale restores nonzero supports and predictive performance. With $\lambda=1$, WGL achieves a mean AUC improvement of $+0.044$ over Logistic Regression ($p=0.0032$), which remains significant after Bonferroni correction. The full λ sensitivity analysis is reported in Appendix D.1. These results show that WGL requires explicit λ calibration, particularly on severely imbalanced datasets and when a continuation schedule is used.

G.6 Optimization Selection in Practice

The comparison above provides practical guidance based on the desired balance between compact feature selection, broader support recovery, computational cost, and tuning requirements. When sparse and interpretable selection is important, VQF provides the most compact solution among the tested methods. It selects approximately 4–5 NCFs per dataset, introduces no optimization-specific hyperparameters beyond the shared penalty C and active-set threshold τ, and achieves the largest improvement over Logistic Regression ($\Delta\text{AUC}=+0.074$, $p=4\times10^{-5}$), which remains significant after Bonferroni correction. It is therefore well suited to applications where the selected indicators should remain compact and auditable.

When broader feature coverage is desired, ADMM-Bisect provides an alternative with approximately 10 selected NCFs. It improves over Logistic Regression by $\Delta\text{AUC}=+0.034$ ($p=0.0124$), which also passes the Bonferroni-corrected threshold, and is approximately $1.3\times$ faster than VQF on the same hardware. ADMM-Bisect introduces one additional hyperparameter, the ADMM penalty ρ, while predictive performance varies by less than 0.010 across $\rho \in \{0.1, 0.5, 1.0, 2.0\}$ in our sensitivity study. We therefore use $\rho=1.0$ as the default in the reported experiments.

WGL provides broader feature selection and direct control over the regularization scale, but requires more careful calibration. With $\lambda=1$, WGL improves over Logistic Regression by $\Delta\text{AUC}=+0.044$ ($p=0.0032$), which remains significant after Bonferroni correction. In contrast, the default high-to-low continuation schedule produces empty supports on all 10 severely imbalanced civil-unrest datasets. We therefore recommend checking the λ sensitivity before applying WGL in highly imbalanced settings rather than relying on the default continuation schedule. The full sensitivity analysis is provided in Appendix D.1, and Table 8 summarizes the main differences among the optimization methods.