Hunting Temporal Bumps in Graphs with Dynamic Vertex Properties: Supplemental Materials

This supplement is available online [1]. The road map of this supplement is as follows. In Section S1, we present the proofs of theorems that are omitted in the main content. In Section S2, we show the feasibility of introducing a regulating weight α into the temporal bump hunting problem. In Section S3, we prove that the static bump hunting problem [2] in NP-hard even when $\alpha = 1$. In Section S4, we show the details of the time complexity of MIRROR. In Section S5, we show the feasibility of introducing a parameter h into S-MIRROR and H-S-MIRROR. In Section S6, we show the feasibility of introducing a parameter b into H-MIRROR and H-S-MIRROR. In Section S7, we compare an existing event detection approach with the temporal bump hunting approach.

S1. THEOREMS AND PROOFS

Here, we present the proofs of theorems that are omitted in the main content.

Theorem 1. *Consider a time interval $T = [t_1, t_m]$; a graph $G(V, E, \eta)$; and a graph $G'(V, E, \xi, c)$. If*

$$\xi^{t_x}(v) = 2\eta^{t_x}(v) \mid \forall v \in V, t_x \in T, \tag{S1}$$

$$c^{t_x}(e) = 1 \mid \forall e \in E, t_x \in T, \tag{S2}$$

then, for any time sub-interval $T_S \subseteq T$; any component $C(V_C, E_C)$ of G; and any tree $\Theta(V_\Theta, E_\Theta)$ of G' that has the same set of vertices with C, i.e., $V_\Theta = V_C$, we have

$$D_C^{T_S} = w_{T_S}(\Theta), \tag{S3}$$

which means that any approximation guarantee (including the optimal guarantee) that holds for maximizing $w_{T_S}(\Theta)$ also holds for maximizing $D_C^{T_S}$, and vise versa.

Proof. First, we have $|V_\Theta| = |E_\Theta| + 1$. Then, we have

$$\begin{aligned} D_C^{T_S} &= p_C^{T_S} - n_C^{T_S} \\ &= p_C^{T_S} - |T_S||V_C| + p_C^{T_S} \\ &= 2\sum_{v \in V_\Theta, t_x \in T_S} \eta^{t_x}(v) - |T_S||E_\Theta| - |T_S| \\ &= \sum_{v \in V_\Theta, t_x \in T_S} \xi^{t_x}(v) - \sum_{e \in E_\Theta, t_x \in T_S} c^{t_x}(v) - |T_S| \\ &= w_{T_S}(\Theta). \end{aligned} \tag{S4}$$

Hence, this theorem holds. □

Theorem 2. *Given a time interval $T = [t_1, t_m]$ and a graph $G'(V, E, \xi, c)$, for any tree $\Theta(V_\Theta, E_\Theta)$ of G' and any time sub-interval $T_i \subseteq T$, we have*

$$UB_{T_i} \geq w_{T_i}(\Theta). \tag{S5}$$

Proof. Let $G'_{T_i}(V, E, w_s, c_s)$ be the aggregated graph built in Line 7 of MIRROR. We have

$$w_{T_i}(\Theta) = \sum_{v \in V_\Theta} w_s(v) - \sum_{e \in E_\Theta} c_s(e) - |T_i|. \tag{S6}$$

We divide V_Θ into two groups V_1 and V_2 such that $w_s(v) > |T_i|$ for every $v \in V_1$, and $w_s(v) \leq |T_i|$ for every $v \in V_2$. Since $c_s(e) \geq |T_i|$ for every edge $e \in E$ and $|E_\Theta| = |V_1| + |V_2| - 1$, we have

$$w_{T_i}(\Theta) \leq \sum_{v \in V_1} w_s(v) + \sum_{v \in V_2} w_s(v) - (|V_1| + |V_2|) \cdot |T_i|. \tag{S7}$$

Suppose that Θ is in a maximum connected component C_x. The maximum value of $\sum_{v \in V_1} w_s(v) - |V_1| \cdot |T_i|$ corresponds to the case where $NUM_{T_i\_C_x} = |V_1|$, *i.e.*, all aggregated vertex prizes in C_x

that are larger than $|T_i|$ are in V_1. This maximum value equals $UB_{T_i\_C_x}$. Moreover, $\sum_{v \in V_2} w_s(v) - |V_2| \cdot |T_i|$ is non-positive. Thus,

$$UB_{T_i} \geq UB_{T_i\_C_x} \geq \sum_{v \in V_1} w_s(v) - |V_1| \cdot |T_i| \geq w_{T_i}(\Theta). \tag{S8}$$

Hence, this theorem holds. □

Theorem 3. MIRROR *has an approximation guarantee of 2 with respect to minimizing $c_{T_S}(\Theta)$ for solving the temporal prize-collecting Steiner tree problem.*

Proof. Suppose that $\{\Theta(V_\Theta, E_\Theta), T_S\}$ is an optimal solution. Let $G'_{T_S}(V, E, w_s, c_s)$ be the aggregated graph of G' during T_S, and let $\Theta_{2T_S}(V_{2T_S}, E_{2T_S})$ be the solution of the Goemans-Williamson approximation scheme [3] for solving the static prize-collecting Steiner tree problem in G'_{T_S}, *i.e.*, if MIRROR conducts Lines 17-18 for T_S, then $\Theta_{2T_S}(V_{2T_S}, E_{2T_S})$ is the pruned tree in Line 18 for T_S. Θ is the optimal solution to the static prize-collecting Steiner tree problem in G'_{T_S}. The Lagrangian-preserving guarantee [4] indicates that

$$\sum_{e \in E_{2T_S}, t_x \in T_S} c^{t_x}(e) + 2 \sum_{v \in V \setminus V_{2T_S}, t_x \in T_S} \xi^{t_x}(v) \leq 2 \sum_{e \in E_\Theta, t_x \in T_S} c^{t_x}(e) + 2 \sum_{v \in V \setminus V_\Theta, t_x \in T_S} \xi^{t_x}(v). \tag{S9}$$

By adding $\sum_{v \in V, t_x \in T \setminus T_S} \xi^{t_x}(v) + |T_S|$ and $2 \sum_{v \in V, t_x \in T \setminus T_S} \xi^{t_x}(v) + 2|T_S|$ to the left and right sides of the above equation, respectively,

$$\begin{aligned} &\sum_{e \in E_{2T_S}, t_x \in T_S} c^{t_x}(e) + 2 \sum_{v \in V \setminus V_{2T_S}, t_x \in T_S} \xi^{t_x}(v) + \sum_{v \in V, t_x \in T \setminus T_S} \xi^{t_x}(v) + |T_S| \quad \leq \\ &2 \sum_{e \in E_\Theta, t_x \in T_S} c^{t_x}(e) + 2 \sum_{v \in V \setminus V_\Theta, t_x \in T_S} \xi^{t_x}(v) + 2 \sum_{v \in V, t_x \in T \setminus T_S} \xi^{t_x}(v) + 2|T_S|, \end{aligned} \tag{S10}$$

Therefore,

$$c_{T_S}(\Theta_{2T_S}) + \sum_{v \in V \setminus V_{2T_S}, t_x \in T_S} \xi^{t_x}(v) \leq 2c_{T_S}(\Theta). \tag{S11}$$

Due to Lines 19-21 of MIRROR, we have

$$c_{T_M}(\Theta_M) \leq c_{T_S}(\Theta_{2T_S}). \tag{S12}$$

$$c_{T_M}(\Theta_M) \leq c_{T_S}(\Theta_{2T_S}) \leq 2c_{T_S}(\Theta) - \sum_{v \in V \setminus V_{2T_S}, t_x \in T_S} \xi^{t_x}(v). \tag{S13}$$

Hence, this theorem holds. □

Theorem 4. *Given a time interval $T = [t_1, t_m]$ and a graph $G'(V, E, \xi, c)$, let $\{\Theta(V_\Theta, E_\Theta), T_S\}$ be an optimal solution to the temporal prize-collecting Steiner tree problem, and let $\{\Theta_M(V_M, E_M), T_M\}$ be the solution of* MIRROR*, then*

$$2w_{T_M}(\Theta_M) + L \geq 2w_{T_S}(\Theta), \tag{S14}$$

where

$$L = \max\{\sum_{e \in E_{2T_i}, t_x \in T_i} c^{t_x}(e) \mid \forall\, T_i \in \Phi\}, \tag{S15}$$

and E_{2T_i} and Φ are in the process of MIRROR *(if Line 18 is not executed for T_i due to the branch and bound process, then we consider $E_{2T_i} = \emptyset$).*

Proof. If Line 18 is skipped for T_S, then $\{\Theta_M(V_M, E_M), T_M\}$ is an optimal solution, and Equation (S14) holds. If Line 18 is not skipped for T_S, then let $\Theta_{2T_S}(V_{2T_S}, E_{2T_S})$ be the pruned tree in Line 18 for time sub-interval T_S. We have

$$w_{T_M}(\Theta_M) \geq w_{T_S}(\Theta_{2T_S}). \tag{S16}$$

By reversing Equation (S9), we have

$$\begin{aligned}&-\sum_{e\in E_{2T_S},t_x\in T_S} c^{t_x}(e) - 2\sum_{v\in V\setminus V_{2T_S},t_x\in T_S} \xi^{t_x}(v) \quad \geq \\ &-2\sum_{e\in E_\Theta,t_x\in T_S} c^{t_x}(e) - 2\sum_{v\in V\setminus V_\Theta,t_x\in T_S} \xi^{t_x}(v).\end{aligned} \tag{S17}$$

By adding $2\sum_{v\in V,t_x\in T_S}\xi^{t_x}(v) - 2|T_S|$ to two sides of the above equation, we have

$$\begin{aligned}&2\sum_{v\in V_{2T_S},t_x\in T_S} \xi^{t_x}(v) - \sum_{e\in E_{2T_S},t_x\in T_S} c^{t_x}(e) - 2|T_S| \geq \\ &2\sum_{v\in V_\Theta,t_x\in T_S} \xi^{t_x}(v) - 2\sum_{e\in E_\Theta,t_x\in T_S} c^{t_x}(e) - 2|T_S|.\end{aligned} \tag{S18}$$

Thus,

$$2w_{T_S}(\Theta_{2T_S}) + \sum_{e\in E_{2T_S},t_x\in T_S} c^{t_x}(e) \geq 2w_{T_S}(\Theta). \tag{S19}$$

Since $L \geq \sum_{e\in E_{2T_S},t_x\in T_S} c^{t_x}(e)$, we have

$$2w_{T_M}(\Theta_M) \geq 2w_{T_S}(\Theta_{2T_S}) \geq 2w_{T_S}(\Theta) - L. \tag{S20}$$

Thus, Equation (S14) still holds. Hence, this theorem holds. □

Lemma 1. *For a time sub-interval $T_S = [t_a, t_c] \subseteq T$, there are two time sub-intervals $[t_a, t_b]$ and $[t_b, t_c]$ in Ω_2 such that $t_a \leq t_b \leq t_c$.*

Proof. There are two possible scenarios as follows. Scenario 1: $|T_S| = 1$, *i.e.*, $t_a = t_c$. Since $\{[t_1,t_2],[t_2,t_3],\cdots,[t_{m-1},t_m]\} \in \Omega_1$, we have $\{[t_1,t_1],[t_2,t_2],\cdots,[t_m,t_m]\} \in \Omega_2$. Thus, $[t_a,t_b]$ and $[t_b,t_c]$ are in Ω_2. Scenario 2: $2 \leq |T_S| \leq m$. There exists $1 \leq i \leq \log_2 m$ such that $2^i \leq |T_S| \leq 2^{i+1}$. As a result, there are time sub-intervals $\{[t_{b-},t_b],[t_b,t_{b+}]\} \in \Omega_1$ such that $t_{b-} \leq t_a \leq t_b \leq t_c \leq t_{b+}$ and

$$b - b_- + 1 = b_+ - b + 1 \leq |T_S| \leq b_+ - b_- + 1.$$

Thus, $[t_a,t_b]$ and $[t_b,t_c]$ are in Ω_2. This lemma holds. □

Theorem 5. *Given a time interval $T = [t_1, t_m]$ and a graph $G'(V,E,\xi,c)$, let $\{\Theta(V_\Theta,E_\Theta),T_S\}$ be an optimal solution to the temporal prize-collecting Steiner tree problem, and let $\{\Theta_{SM}(V_{SM},E_{SM}),T_{SM}\}$ be the solution of* S-MIRROR, *then*

$$2w_{T_{SM}}(\Theta_{SM}) + H + 1 \geq w_{T_S}(\Theta), \tag{S21}$$

$$2c_{T_{SM}}(\Theta_{SM}) - H - 1 \leq \sum_{v\in V,t_x\in T} \xi^{t_x}(v) + c_{T_S}(\Theta), \tag{S22}$$

where

$$H = \max\{\sum_{e\in E_\Theta} c^{t_b}(e) - \sum_{v\in V_\Theta} \xi^{t_b}(v) \mid \forall t_b \in T\} + \max\{\kappa_1,\kappa_2\}, \tag{S23}$$

where κ_1 is the maximum value of $\sum_{e\in E_{2T_i},t_x\in T_i} c^{t_x}(e)$ for such $T_i \in \Omega_2$ that Line 18 of MIRROR *is executed, and κ_2 is the maximum value of ζ_{T_i} for such $T_i \in \Omega_2$ that Line 12 or 15 of* MIRROR *is executed.*

Proof. Let $T_S = [t_a, t_c]$. By Lemma 1, there are $T_1 = [t_a, t_b]$ and $T_2 = [t_b, t_c]$ in Ω_2, where $t_a \leq t_b \leq t_c$. Since $|T_S| = |T_1| + |T_2| - 1$,

$$\begin{aligned}w_{T_S}(\Theta) = &\sum_{v\in V_\Theta,t_x\in T_1} \xi^{t_x}(v) - \sum_{e\in E_\Theta,t_x\in T_1} c^{t_x}(e) + \sum_{v\in V_\Theta,t_x\in T_2} \xi^{t_x}(v) - \sum_{e\in E_\Theta,t_x\in T_2} c^{t_x}(e) \\ &- \sum_{v\in V_\Theta} \xi^{t_b}(v) + \sum_{e\in E_\Theta} c^{t_b}(e) - |T_1| - |T_2| + 1.\end{aligned} \tag{S24}$$

Let $\Theta_{T_1}(V_{T_1}, E_{T_1})$ and $\Theta_{T_2}(V_{T_2}, E_{T_2})$ be the optimal solutions to the static prize-collecting Steiner tree problem in the aggregated graphs during T_1 and T_2 (*i.e.*, G'_{T_1} and G'_{T_2}), respectively. Let $\Theta_{2T_1}(V_{2T_1}, E_{2T_1})$ and $\Theta_{2T_2}(V_{2T_2}, E_{2T_2})$ be the solutions of the Goemans-Williamson approximation scheme to the static prize-collecting Steiner tree problem in G'_{T_1} and G'_{T_2}, respectively. By the Lagrangian-preserving guarantee [4], we have

$$\sum_{e\in E_{2T_1},t_x\in T_1} c^{t_x}(e) + 2\sum_{v\in V\setminus V_{2T_1},t_x\in T_1} \xi^{t_x}(v) \le 2\sum_{e\in E_{T_1},t_x\in T_1} c^{t_x}(e) + 2\sum_{v\in V\setminus V_{T_1},t_x\in T_1} \xi^{t_x}(v), \tag{S25}$$

$$\sum_{e\in E_{2T_2},t_x\in T_2} c^{t_x}(e) + 2\sum_{v\in V\setminus V_{2T_2},t_x\in T_2} \xi^{t_x}(v) \le 2\sum_{e\in E_{T_2},t_x\in T_2} c^{t_x}(e) + 2\sum_{v\in V\setminus V_{T_2},t_x\in T_2} \xi^{t_x}(v). \tag{S26}$$

By adding $2\sum_{v\in V,t_x\in T_1} \xi^{t_x}(v)$ and $2\sum_{v\in V,t_x\in T_2} \xi^{t_x}(v)$ to both sides of the reverse of the above two equations respectively, we have

$$\begin{aligned} &2\sum_{v\in V_{2T_1},t_x\in T_1} \xi^{t_x}(v) - \sum_{e\in E_{2T_1},t_x\in T_1} c^{t_x}(e) \ge \\ &2\sum_{v\in V_{T_1},t_x\in T_1} \xi^{t_x}(v) - 2\sum_{e\in E_{T_1},t_x\in T_1} c^{t_x}(e) \ge \\ &2\sum_{v\in V_{\Theta},t_x\in T_1} \xi^{t_x}(v) - 2\sum_{e\in E_{\Theta},t_x\in T_1} c^{t_x}(e), \end{aligned} \tag{S27}$$

$$\begin{aligned} &2\sum_{v\in V_{2T_2},t_x\in T_2} \xi^{t_x}(v) - \sum_{e\in E_{2T_2},t_x\in T_2} c^{t_x}(e) \ge \\ &2\sum_{v\in V_{T_2},t_x\in T_2} \xi^{t_x}(v) - 2\sum_{e\in E_{T_2},t_x\in T_2} c^{t_x}(e) \ge \\ &2\sum_{v\in V_{\Theta},t_x\in T_2} \xi^{t_x}(v) - 2\sum_{e\in E_{\Theta},t_x\in T_2} c^{t_x}(e). \end{aligned} \tag{S28}$$

By combining Equation (S24) and the above twos, we have

$$\begin{aligned} &2\sum_{v\in V_{2T_1},t_x\in T_1} \xi^{t_x}(v) - \sum_{e\in E_{2T_1},t_x\in T_1} c^{t_x}(e) \\ &+2\sum_{v\in V_{2T_2},t_x\in T_2} \xi^{t_x}(v) - \sum_{e\in E_{2T_2},t_x\in T_2} c^{t_x}(e) \\ &-2\sum_{v\in V_{\Theta}} \xi^{t_b}(v) + 2\sum_{e\in E_{\Theta}} c^{t_b}(e) - 2|T_1| - 2|T_2| + 2 \quad \ge 2w_{T_S}(\Theta). \end{aligned} \tag{S29}$$

For each $T_i \in \{T_1, T_2\}$, since

$$w_{T_{SM}}(\Theta_{SM}) \ge \sum_{v\in V_{2T_i},t_x\in T_i} \xi^{t_x}(v) - \sum_{e\in E_{2T_i},t_x\in T_i} c^{t_x}(e) - |T_i|, \tag{S30}$$

we further have

$$\begin{aligned} 4w_{T_{SM}}(\Theta_{SM}) + \sum_{e\in E_{2T_1},t_x\in T_1} c^{t_x}(e) + \sum_{e\in E_{2T_2},t_x\in T_2} c^{t_x}(e) + 2\sum_{e\in E_{\Theta}} c^{t_b}(e) \\ -2\sum_{v\in V_{\Theta}} \xi^{t_b}(v) + 2 \quad \ge 2w_{T_S}(\Theta). \end{aligned} \tag{S31}$$

For each $T_i \in \{T_1, T_2\}$, if Line 18 of MIRROR is executed, then,

$$\kappa_1 \ge \sum_{e\in E_{2T_i},t_x\in T_i} c^{t_x}(e). \tag{S32}$$

If Line 18 of MIRROR is not executed, then,

$$\kappa_2 \ge \zeta_{T_i} \ge \sum_{e\in E_{2T_i},t_x\in T_i} c^{t_x}(e). \tag{S33}$$

Therefore, for each $T_i \in \{T_1, T_2\}$,

$$H \geq \sum_{e \in E_{2T_i}, t_x \in T_i} c^{t_x}(e) + \sum_{e \in E_\Theta} c^{t_b}(e) - \sum_{v \in V_\Theta} \xi^{t_b}(v). \tag{S34}$$

Thus, Equation (S21) can be deducted from Equation (S31). Equation (S22) can be deducted by changing solution weights in Equation (S21) to solution costs. Hence, this theorem holds. □

Theorem 6. *Given a time interval $T = [t_1, t_m]$ and a graph $G'(V, E, \xi, c)$ built via Theorem 1, let $\{\Theta(V_\Theta, E_\Theta), T_S\}$ be an optimal solution to the temporal prize-collecting Steiner tree problem. If there is at least one positive vertex prize in G' during T, then*

$$|E_\Theta| \leq \min\{|V| - 1, 2|V_{pos}| - 1 - \frac{1}{|T|}\}, \tag{S35}$$

where V_{pos} is the set of vertices that have at least one positive prize during T, i.e., $V_{pos} = \{v \mid \forall v \in V, \sum_{t_x \in T} \xi^{t_x}(v) > 0\}$.

Proof. Since $|V_\Theta| \leq |V|$, $|E_\Theta| \leq |V| - 1$. We prove that $|E_\Theta| \leq 2|V_{pos}| - 1 - \frac{1}{|T|}$ as follows. Since G' is built via Theorem 1, there is a vertex $v \in V$ and a time slot $t_x \in T$ such that $\xi^{t_x}(v) = 2$. Then, there is a tree $\Theta' = \{v\}$ and a time sub-interval $T' = [t_x, t_x]$, and

$$w_{T_S}(\Theta) \geq w_{T'}(\Theta') = 1. \tag{S36}$$

Since each vertex prize in G' during T is either 0 or 2,

$$\sum_{v \in V_\Theta, t_x \in T_S} \xi^{t_x}(v) \leq |T_S| \cdot 2 \cdot |V_{pos}|. \tag{S37}$$

Thus,

$$|T_S| \cdot 2 \cdot |V_{pos}| - |E_\Theta| \cdot |T_S| - |T_S| \geq w_{T_S}(\Theta) \geq 1, \tag{S38}$$

$$|E_\Theta| \leq 2|V_{pos}| - 1 - \frac{1}{|T|}. \tag{S39}$$

Hence, this theorem holds. □

Theorem 7. *Given a time interval $T = [t_1, t_m]$ and a graph $G'(V, E, \xi, c)$ built via Theorem 1, if there is at least one positive vertex prize in G' during T, then* H-MIRROR *and* H-S-MIRROR*, as well as* MIRROR *and* S-MIRROR*, have a trivial approximation guarantee of $\frac{1}{m|V|}$ with respect to maximizing $w_{T_S}(\Theta)$.*

Proof. Let $\{\Theta_a(V_{\Theta_a}, E_{\Theta_a}), T_a\}$ be the solution of H-MIRROR, or H-S-MIRROR, or MIRROR, or S-MIRROR. Let $\{\Theta(V_\Theta, E_\Theta), T_S\}$ be an optimal solution. Since G' is built via Theorem 1, there is a vertex $v \in V$ and a time slot $t_x \in T$ such that $\xi^{t_x}(v) = 2$. Then, there is a tree $\Theta' = \{v\}$ and a time sub-interval $T' = [t_x, t_x]$. The Goemans-Williamson approximation scheme guarantees that $w_{T_a}(\Theta_a) \geq w_{T'}(\Theta') = 1$. Moreover, we have

$$w_{T_S}(\Theta) \leq 2m|V| - m(|V| - 1) - m \leq m|V| \cdot w_{T_a}(\Theta_a). \tag{S40}$$

Hence, this theorem holds. □

S2. INTRODUCING THE REGULATING WEIGHT α

As discussed in the main content, like the previous work [2], we can add a regulating weight $\alpha > 0$ into the temporal bump hunting problem, *i.e.*, we define the temporal discrepancy as

$$D_C^{T_S} = \alpha p_C^{T_S} - n_C^{T_S}. \tag{S41}$$

We need to conduct some changes on the main content for incorporating α into the problem setting. The details are as follow. First, we need to change Equation (S1) in Theorem 1 to

$$\xi^{t_x}(v) = (\alpha + 1)\eta^{t_x}(v) \mid \forall v \in V, t_x \in T. \tag{S42}$$

The above proof of Theorem 1 can be easily modified for incorporating α. Then, we need to change Equation (S35) in Theorem 6 to

$$|E_\Theta| \leq \min\{|V| - 1, (1 + \alpha) \cdot |V_{pos}| - 1 - \frac{\alpha}{|T|}\}. \tag{S43}$$

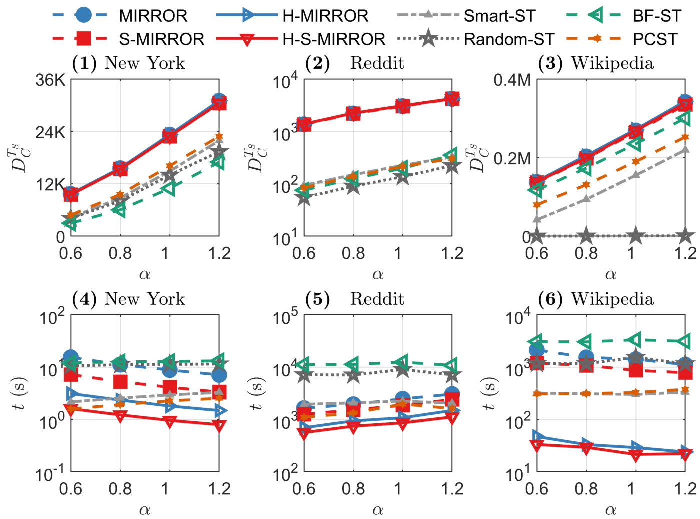

Fig. S1. Experiment results of varying α.

The above proof of Theorem 6 can also be easily modified for incorporating α. Based on the above equation, when G' is built via Theorem 1 (with α), we have

$$\sum_{e \in E_\Theta} c^{t_b}(e) - \sum_{v \in V_\Theta} \xi^{t_b}(v) \leq \min\{|V| - 1, (1+\alpha) \cdot |V_{pos}| - 1 - \frac{\alpha}{|T|}\}. \tag{S44}$$

Then, by Equation (S21), the upper bound of $w_{T_S}(\Theta)$ produced using the solution of S-MIRROR is

$$2w_{T_{SM}}(\Theta_{SM}) + \max\{\kappa_1, \kappa_2\} + \min\{|V|, (1+\alpha) \cdot |V_{pos}| - \frac{\alpha}{|T|}\}.$$

With these changes, we can introduce α into the problem setting. We show the experiment results of varying α in Figure S1. We observe that $D_C^{T_S}$ generally increases with α, since queried vertices are more valued when α is large (see Equation (S41)). In Figures S1 (4) and (6), t values of the proposed algorithms decrease with α. We explain this as follows. Queried vertices in the New York and Wikipedia graphs are often close to each other. As α increases, vertex prizes increase, and the above algorithms find solutions that include more queried vertices and span larger time sub-intervals. These algorithms use these solutions to prune more time sub-intervals in the branch and bound process (*i.e.*, Lines 8-16 in MIRROR). As a result, the running times of these algorithms decrease with α. In comparison, in Figure S1 (5), t values of these algorithms increase with α. The reason is that queried vertices in the Reddit graph are often far away from each other, which means that, as α increases, these algorithms may not prune more time sub-intervals in the branch and bound process. At the same time, the growing and pruning processes (*i.e.*, Lines 17-18 in MIRROR) become slower, since vertex prizes increase with α, and as a result more not queried vertices are included in the growing process and removed in the pruning process.

S3. THE STATIC BUMP HUNTING PROBLEM IS NP-HARD EVEN WHEN $\alpha = 1$

In the static bump hunting problem [2], there is a graph $G(V, E)$, and a set of queried vertices $Q \subseteq V$. Let $C(V_C, E_C)$ be a component of G. We refer to p_C as the number of queried vertices in C, i.e., $p_C = |V_C \cap Q|$. We refer to n_C as the number of not queried vertices in C, i.e., $n_C = |V_C \setminus Q|$. Then, the static discrepancy of C in [2] is

$$g(C) = \alpha p_C - n_C \tag{S45}$$

The static bump hunting problem [2] is to find a component C such that $g(C)$ is maximized.

The previous work [2] proves the NP-hardness of the static bump hunting problem without the restriction of α via a transformation of the set cover problem [5]. We modify this transformation, and prove that the static bump hunting problem is NP-hard even when α is restricted to 1 as follows.

Theorem 8. *The static bump hunting problem [2] is NP-hard when $\alpha = 1$.*

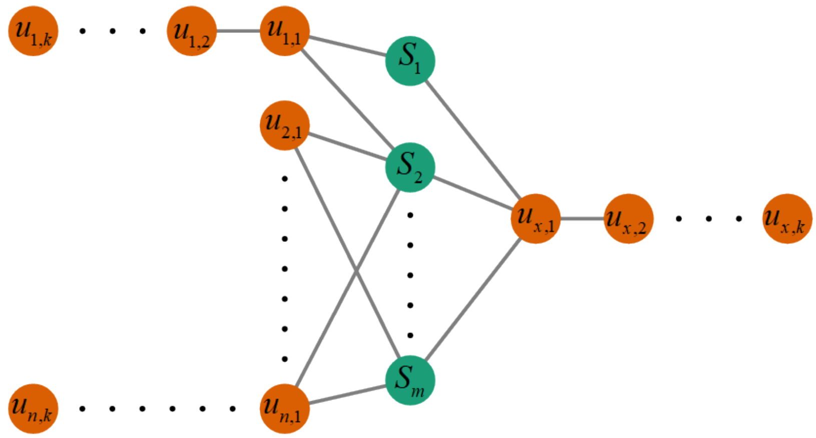

Fig. S2. The static bump hunting problem is NP-hard when $\alpha = 1$.

Proof. In the set cover problem, there is a ground set $U = \{u_1, \cdots, u_n\}$ of n elements, a collection of $C = \{S_1, \cdots, S_m\}$ of m sub-sets of U, and an integer k. The decision question of the set cover problem is whether there are at most k sets in C whose union contains all the elements in the ground set U. Given an instance of the set cover problem, we create a graph G, *i.e.*, an instance of the static bump hunting problem, in Figure S2. There are $(n+1)k+m$ vertices in total, k concatenated vertices for each element u_i (*e.g.*, $\{u_{1,1}, \cdots, u_{1,k}\}$ for u_1), one vertex for each set S_j, and k additional concatenated vertices $\{u_{x,1}, \cdots, u_{x,k}\}$. There is an edge $(u_{i,1}, S_j)$ if and only if $u_i \in S_j$ for every $i \in [1,n]$ and $j \in [1,m]$, and m additional edges $(S_j, u_{x,1})$ for every $j \in [1,m]$. In Figure S2, we consider orange and green vertices as queried and not queried vertices, respectively.

We set $\alpha = 1$ in the static bump hunting problem. A decision question of the static bump hunting problem is whether there is a component C of G that has a discrepancy $g(C) \geq nk$. If such a C exists, then $\{u_{1,1}, \cdots, u_{n,1}, u_{x,1}\}$ should be in C, otherwise $g(C) < nk$. Moreover, there are at most k not queried vertices in C, otherwise $g(C) < (n+1)k - k = nk$. Thus, if there is a component C of G that has a discrepancy $g(C) \geq nk$, then there are at most k sets in C whose union contains all the elements in the ground set U. That is to say, answering the decision question of the static bump hunting problem is equivalent to answering the decision question of the set cover problem. Since the set cover problem is NP-hard, the static bump hunting problem [2] is NP-hard when $\alpha = 1$. Hence, this theorem holds. $\square$

The static bump hunting problem [2] with the restriction of $\alpha = 1$ is a special case of the temporal bump hunting problem where $m = 1$. Thus, we have the following corollary.

Corollary 1. *The temporal bump hunting problem is NP-hard.*

S4. THE TIME COMPLEXITY OF MIRROR

The time complexity of MIRROR is

$$O\Big(m^2|V| + m^2 d|E| \log |V| + m^2|\cup v_{pos}| + m^3\Big),$$

where d is the precision of vertex prizes and edge costs (details in [6]), and $|\cup v_{pos}|$ is the number of positive vertex prizes, which is at most $m|V|$. The details of the above time complexity are as follows.

Initially, MIRROR sorts all time sub-intervals from large to small (Line 1 of MIRROR), which induces a cost of $O(m^2 \log m)$, since there are $m(m+1)/2$ time sub-intervals in total. Then, it does the initialization (Line 2) in $O(1)$ time. We use adjacency lists based on hashes to store graphs. As a result, reducing G' via Degree-0-1-2 test (Line 3) takes $O(|V|)$ time. It conducts a depth-first search to mark maximum connected components of G' (Line 4) at a cost of $O(|V|+|E|)$.

After that, MIRROR enumerates $O(m^2)$ time sub-intervals using a loop (Line 5). For each time sub-interval $T_i \subseteq T$, it checks whether T_i is in the hash table P or not (Line 6) in $O(1)$ time. If T_i is not in P, it builds G'_{T_i} (Line 7), which induces a cost of $O(|V|+|E|+|\cup v_{pos}|)$, since it aggregates all vertex prizes and enumerates all edges for updating w_s and c_s (note that, updating c_s takes $O(|E|)$ time, since all edge costs in Theorem 1 equal 1; and updating w_s takes $O(|V|+|\cup v_{pos}|)$

time, since we only associate positive, but not zero, prizes with vertices). Then, it computes ζ_{T_i} in $O(|V|)$ time, and checks two conditions (Lines 8-9) in $O(1)$ time.

If $\zeta_{T_i} \leq w_{T_M}(\Theta_M)$ (Line 9), it inserts all time sub-intervals in T_i into P (Line 10), which has a cost of $O(m^3)$ throughout the loop. The reason is as follows. Assume that it prunes time sub-intervals under the above condition for every time sub-interval T_i such that $|T_i| = x$ and $1 \leq x \leq m$. This means that it does not do this for any other time sub-interval that has a different length. The number of time sub-intervals with a length of x is $m - x + 1$. The number of time sub-intervals that are contained by a time sub-interval with a length of x is $x(x+1)/2$. Thus, the number of times that we insert time sub-intervals into P is $f(x) = (m - x + 1) \cdot x(x+1)/2$. Since $O(f(x))$ is always $O(m^3)$, the process of pruning time sub-intervals in T_i (Line 10) induces a cost of $O(m^3)$ throughout the loop.

MIRROR computes UB_{T_i} in $O(|V|)$ time, and checks whether $UB_{T_i} \leq w_{T_M}(\Theta_M)$ (Line 14) in $O(1)$ time. If $UB_{T_i} > w_{T_M}(\Theta_M)$, it produces a raw tree Θ_{1T_i} (Line 17) in $O(d|E|\log|V|)$ time [6]. It further prunes this raw tree as Θ_{2T_i} (Line 18), which induces a cost of $O(|V|)$ [7]. It computes $w_{T_i}(\Theta_{2T_i})$ via $w_{G'_{T_i}}(\Theta_{2T_i})$ using Equation (13) in the main content, which has a cost of $O(|V|)$. If $w_{T_i}(\Theta_{2T_i}) > w_{T_M}(\Theta_M)$ (Line 19), it updates Θ_M and T_M (Line 20) at a cost of $O(|V|)$. It returns Θ_M and T_M (Line 24), which induces a cost of $O(|V|)$.

S5. INTRODUCING A PARAMETER h INTO S-MIRROR AND H-S-MIRROR

As discussed in the main content, we could add a parameter $h \in \mathbb{N}$ into Ω_3 in Line 1 of S-MIRROR, *i.e.*, to let Ω_3 contain $\min\{h \cdot m\log_2 m, |\Upsilon|\}$ time sub-intervals that are selected from Υ uniformly at random. Here, we add the parameter h into S-MIRROR and H-S-MIRROR, and conduct experiments of varying h.

First, we note that, after adding h, the number of selected time sub-intervals in Line 1 of S-MIRROR is

$$O\Big(|\Omega|\Big) = O\Big(\min\{h, \frac{m}{\log m}\} \cdot m\log m\Big). \tag{S46}$$

Then, the time complexity of S-MIRROR is

$$O\Big(\min\{h, \frac{m}{\log m}\} \cdot (m\log m \cdot |V| + m\log m \cdot d|E|\log|V| + m\log m \cdot |\cup v_{pos}|) + m^3\Big),$$

where $|\cup v_{pos}|$ is the number of positive vertex prizes and is at most $m|V|$. As discussed in the main content, we often have $|\cup v_{pos}| \ll m|V|$ in practice. As a result, by setting h to a small value, the cost of S-MIRROR with respect to m is nearly linear in practice.

In particular, we conduct experiments of varying h, and show the results in Figure S3. The default settings of parameters are the same with the experiments in the main content. It can be seen that, for S-MIRROR and H-S-MIRROR, $D_C^{T_S}$ and t often increase with h, since these two

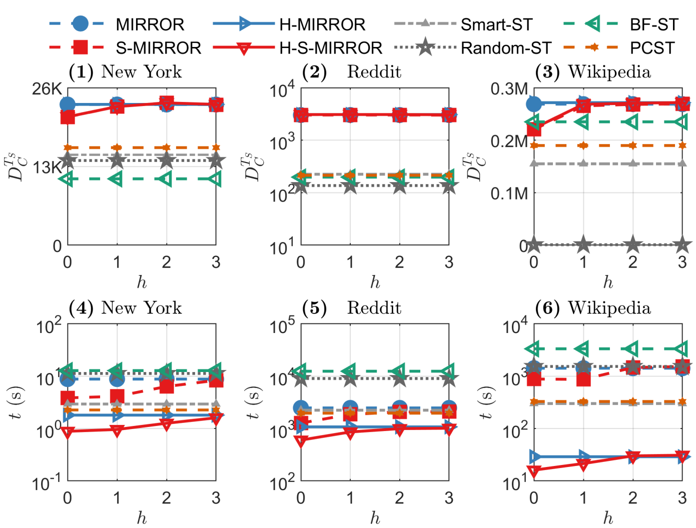

Fig. S3. Experiment results of varying h.

algorithms compute more time sub-intervals as h increases. Specifically, for S-MIRROR and H-S-MIRROR, $D_C^{T_S}$ values do not increase much when increasing h from 1 to larger values, while t values may still increase when increasing h from 1 to larger values, *e.g.*, Figure S3 (4). Thus, setting $h = 1$ gives S-MIRROR and H-S-MIRROR a high performance. As a result, for the easy use of S-MIRROR and H-S-MIRROR, we do not introduce h in the main content.

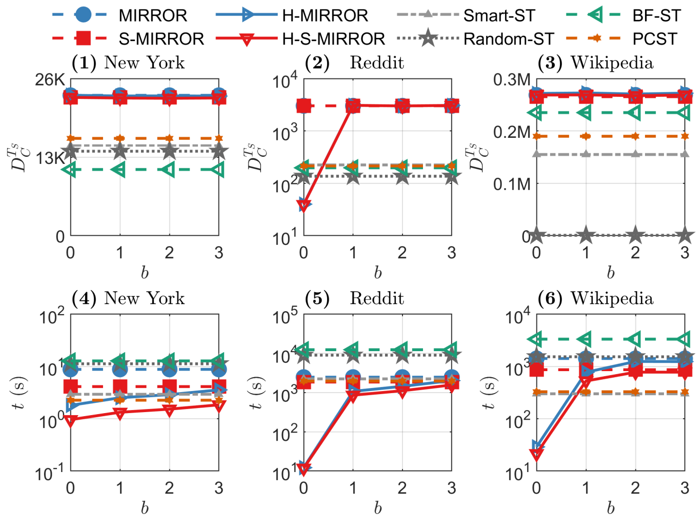

Fig. S4. Experiment results of varying b.

S6. INTRODUCING A PARAMETER b INTO H-MIRROR AND H-S-MIRROR

As discussed in Section 5.2 in the main content, we can treat b in H-MIRROR and H-S-MIRROR as a parameter. We do this in this section, and present the experiment results of varying b in Figure S4. We observe that, in Figures S4 (4-6), t often increases with b for H-MIRROR and H-S-MIRROR, since these two algorithms perform breadth first searches from queried vertices with a maximum depth of b, and then compute all the searched vertices. We further note that, for New York and Wikipedia in Figures S4 (1) and (3), $D_C^{T_S}$ does not change much with b. This shows that temporal bumps in New York and Wikipedia mostly contain queried vertices. In comparison, for Reddit in Figure S4 (2), $D_C^{T_S}$ increases significantly when b increases from 0 to 1. The reason is as follows. There are three types of vertices in the Reddit graph: vertices representing (i) communities, (ii) keywords, and (iii) pairs of communities and keywords. The third type of vertices may be queried, and are linked to the first and second types of vertices, which are not queried. As a result, when b is smaller than 1, queried vertices are often not connected with each other in the sub-graph constructed by the breadth first searched vertices, and when b is larger than or equal to 1, queried vertices are often connected with each other in the above sub-graph. These experiment results show that H-MIRROR and H-S-MIRROR have a sufficiently high performance when defining b as the minimum possible number of vertices between two queried vertices (*i.e.*, $b = 0$ for New York and Wikipedia, and $b = 1$ for Reddit). Thus, in the main content, we define b in the above way, but do not treat b as a parameter, for achieving the easy use of H-MIRROR and H-S-MIRROR.

S7. COMPARING AN EXISTING EVENT DETECTION APPROACH

In the main content, we use the proposed H-MIRROR to detect a cluster of closely related Wikipedia pages and a time sub-interval that correspond to our intensive attention to the US-Iran conflict on 3rd January 2020. As discussed in the Introduction section, both the graph and the temporal information are essential in this case, and the existing work on event detection does not suit this case, since most such work focuses on non-graph-structured datasets (*e.g.*, [8–10]), while the other work that focuses on graph-structured datasets either does not consider the temporal information (*e.g.*, [11–13]), or only suits edge-evolving graphs where event-related activities are associated with edges (*e.g.*, [14–16]). The above work that focuses on non-graph-structured datasets or edge-evolving graphs cannot be applied to the Wikipedia case. Differently, the above work that focuses on static graphs can be applied to the Wikipedia case. Here, we apply such a work in [11] to the Wikipedia case, for showing the unique usefulness of our work.

Table S1. The results of EventTree-PD [11] for the Wikipedia data on 3rd Jan. 2020.

λ	Wikipedia pages
0.00006	Iran–Iraq War, Shia Islam, Ruhollah Khomeini, Cold War, Qasem Soleimani, World War III, Muhammad, Iran, Ali Khamenei
0.00008	New York City, Carlos Ghosn, Barack Obama, Ruhollah Khomeini, United Kingdom, Lockheed Martin F-35 Lightning II, Sark, List of countries by GDP (nominal), Will Smith, Philippines, Juan Sebastián Elcano, Osama bin Laden, Hezbollah, Conscription in the United States, Saudi Arabia, Iran, Japan, General Atomics MQ-9 Reaper, World War II, Qasem Soleimani, Shia Islam, Iran–Iraq War, Iranian Revolution, Iran hostage crisis, Cold War, Google, Vietnam War, Netflix, Ali Khamenei, Selective Service System, Mahmoud Ahmadinejad, Cameron Diaz, Benji Madden, Russia, Star Wars, Brooklyn, Iran–United States relations, Artificial intelligence, China, Adam Sandler, World War III, Muhammad, Ali Mohamed, The Fresh Prince of Bel-Air, Nissan, Ed Rendell, West Nickel Mines School shooting, Jeffrey B. Miller

Table S2. The results of EventTree-PD [11] for the Wikipedia data in each hour on 3rd Jan. 2020.

Time span	Wikipedia pages
00:00-01:00 3rd Jan. 2020	West Nickel Mines School shooting
01:00-02:00 3rd Jan. 2020	West Nickel Mines School shooting
02:00-03:00 3rd Jan. 2020	Qasem Soleimani
03:00-04:00 3rd Jan. 2020	Qasem Soleimani
04:00-05:00 3rd Jan. 2020	Qasem Soleimani
05:00-06:00 3rd Jan. 2020	Iran, Jerusalem, Qasem Soleimani, Ali Khamenei, Bible
06:00-07:00 3rd Jan. 2020	Qasem Soleimani
07:00-08:00 3rd Jan. 2020	Qasem Soleimani
08:00-09:00 3rd Jan. 2020	Iran, Jerusalem, Qasem Soleimani, Ali Khamenei, Bible
09:00-10:00 3rd Jan. 2020	Iran, Puerto Rico, Ricky Martin, Qasem Soleimani, List of countries by GDP (nominal), Ali Khamenei, Almaty, Tehran
10:00-11:00 3rd Jan. 2020	Puerto Rico, Ricky Martin, List of countries by GDP (nominal), Qasem Soleimani, Iran, Ali Khamenei
11:00-12:00 3rd Jan. 2020	Qasem Soleimani, Iran, Ali Khamenei
12:00-13:00 3rd Jan. 2020	Ali Khamenei, Qasem Soleimani, Iran, Philippines, Juan Sebastián Elcano
13:00-14:00 3rd Jan. 2020	Ali Khamenei, Qasem Soleimani, Iran
14:00-15:00 3rd Jan. 2020	Iran–United States relations, Ruhollah Khomeini, Iran–Iraq War, Qasem Soleimani, Iran, Ali Khamenei
15:00-16:00 3rd Jan. 2020	Qasem Soleimani, World War III, Iran, Ruhollah Khomeini, Cold War, Ali Khamenei
16:00-17:00 3rd Jan. 2020	Ruhollah Khomeini, Iran–Iraq War, Qasem Soleimani, World War III, Iran, Ali Khamenei, India, Cold War
17:00-18:00 3rd Jan. 2020	Ali Khamenei, Qasem Soleimani, Ruhollah Khomeini, Iran–Iraq War, Cold War, Iran–United States relations, World War III, Iran
18:00-19:00 3rd Jan. 2020	Ruhollah Khomeini, Iran–Iraq War, Cold War, Qasem Soleimani, World War III, Iran, Ali Khamenei
19:00-20:00 3rd Jan. 2020	Qasem Soleimani, Iran, Iran–United States relations, Ruhollah Khomeini, Iran–Iraq War, Ali Khamenei
20:00-21:00 3rd Jan. 2020	Iran, Conscription in the United States, Ruhollah Khomeini, Iran–Iraq War, United States Armed Forces, Martin Scorsese, Cameron Diaz, New York City, Cold War, Benji Madden, Gulf War, Selective Service System, Ali Khamenei, Qasem Soleimani, Brooklyn, World War III, Supreme Court of the United States
21:00-22:00 3rd Jan. 2020	Adam Driver, Ali Khamenei, Cameron Diaz, Qasem Soleimani, Iran–Iraq War, Shia Islam, Ruhollah Khomeini, Barack Obama, Iran, Iran–United States relations, Benji Madden, Cold War, Robin Williams, John Travolta, World War III, Muhammad
22:00-23:00 3rd Jan. 2020	Conscription in the United States, Iran, Kurds, Ruhollah Khomeini, Shia Islam, Iran–Iraq War, Cameron Diaz, Qasem Soleimani, Selective Service System, Ali Khamenei, Martin Scorsese, New York City, Cold War, Benji Madden, Iranian Revolution, Vietnam War, Brooklyn, World War III, Muhammad, Supreme Court of the United States
23:00-24:00 3rd Jan. 2020	Iran, Iran–United States relations, Ali Khamenei, Qasem Soleimani, Iran–Iraq War, Shia Islam, Ruhollah Khomeini, Cold War, World War III, Muhammad

Specifically, we apply the EventTree-PD technique in [11]. This technique inputs a static graph $G(V, E, w, c)$, where w is a function that maps each vertex $v \in V$ to a non-negative weight value $w(v)$, and c is a function that maps each edge $e \in E$ to a non-negative cost value $c(e)$. It outputs a tree Θ that maximizes $\lambda \cdot W(\Theta) - C(\Theta)$, where $\lambda > 0$ is a regulating weight, $W(\Theta)$ is the sum of vertex weights in Θ, and $C(\Theta)$ is the sum of edge costs in Θ. EventTree-PD in [11] is similar to the baseline algorithm PCST [2] in the main content, since both techniques input a static graph and output a tree by solving the static prize-collecting Steiner tree problem [17]. The difference between these two techniques is that PCST maximizes the static discrepancy of the tree.

We apply EventTree-PD to the Wikipedia data on 3rd January 2020. In particular, we apply EventTree-PD to the Wikipedia graph where each vertex weight is the number of times that the corresponding page has been viewed on 3rd January 2020, and each edge cost is 1. We use two different settings of λ, and show the Wikipedia pages in the trees returned by EventTree-PD in Table S1. The sizes of returned trees increase with λ, since a larger value of λ weights vertices more. When $\lambda = 0.00006$, EventTree-PD detects a cluster of pages related to the US-Iran conflict. This shows that people are paying intensive attention to the conflict on 3rd January 2020. In comparison, we may not be able to discover this knowledge when $\lambda = 0.00008$, since many detected pages are unrelated to the conflict when $\lambda = 0.00008$. Nevertheless, due to the neglect of the temporal information, EventTree-PD cannot mine the temporal knowledge like H-MIRROR even when $\lambda = 0.00006$, such as the temporal knowledge that when our intensive attention to the US-Iran conflict starts, or whether our intensive attention to the US-Iran conflict is ongoing to the end of 3rd Jan. 2020. Differently, H-MIRROR can mine such temporal knowledge by hunting the temporal bump in Figure 1 in the main content.

A possible solution to the above shortage of EventTree-PD is to apply it to the Wikipedia data in every hour on 3rd January 2020. We show the results of doing this in Table S2, where $\lambda = 0.00006$. Unlike Table S1, we do not set $\lambda = 0.00008$ here, since the pages detected by EventTree-PD are too many to be listed when $\lambda = 0.00008$. We observe that EventTree-PD detects different clusters of pages in different hours, and the detected pages in different hours often vary a lot. Since many detected pages are unrelated to the US-Iran conflict (*e.g.*, the pages of Jerusalem and Bible in the hour of 05:00-06:00, and the pages of New York City and Brooklyn in the hour of 22:00-23:00), EventTree-PD cannot detect a cluster of closely related Wikipedia pages and a time sub-interval that correspond to our consistent and intensive attention to the US-Iran conflict. In comparison, the proposed H-MIRROR can meet this challenge, as illustrated in Figure 1 in the main content. This shows the particular usefulness of the proposed temporal bump hunting approach for analyzing graph-structured datasets with dynamic vertex properties.

REFERENCES FOR THE SUPPLEMENT

1. "Supplement," (2021). https://github.com/rucdatascience/temporal_bh/blob/main/Supplement.pdf.
2. A. Gionis, M. Mathioudakis, and A. Ukkonen, "Bump hunting in the dark: Local discrepancy maximization on graphs," IEEE Transactions on Knowl. Data Eng. **29**, 529–542 (2017).
3. M. X. Goemans and D. P. Williamson, "A general approximation technique for constrained forest problems," SIAM Journal on Computing **24**, 296–317 (1995).
4. F. A. Chudak, T. Roughgarden, and D. P. Williamson, "Approximate k-MSTs and k-Steiner trees via the primal-dual method and lagrangean relaxation," Math. Program. **100**, 411–421 (2004).
5. R. M. Karp, "Reducibility among combinatorial problems," in *Complexity of computer computations,* (Springer, 1972), pp. 85–103.
6. C. Hegde, P. Indyk, and L. Schmidt, "A fast, adaptive variant of the Goemans-Williamson scheme for the prize-collecting Steiner tree problem," in *Workshop of the 11th DIMACS Implementation Challenge,* (2014).
7. Y. Sun, M. Brazil, D. Thomas, and S. Halgamuge, "The fast heuristic algorithms and post-processing techniques to design large and low-cost communication networks," IEEE/ACM Transactions on Netw. **27**, 375–388 (2019).
8. M. Fedoryszak, B. Frederick, V. Rajaram, and C. Zhong, "Real-time event detection on social data streams," in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery and data mining,* (2019), pp. 2774–2782.
9. A. Perer and F. Wang, "Frequence: Interactive mining and visualization of temporal frequent event sequences," in *Proceedings of the 19th international conference on Intelligent User Interfaces,* (2014), pp. 153–162.
10. X. Ao, H. Shi, J. Wang, L. Zuo, H. Li, and Q. He, "Large-scale frequent episode mining from complex event sequences with hierarchies," ACM Transactions on Intell. Syst. Technol. **10**, 1–26 (2019).
11. P. Rozenshtein, A. Anagnostopoulos, A. Gionis, and N. Tatti, "Event detection in activity networks," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining,* (2014), pp. 1176–1185.
12. Y. Liu, B. Zhou, F. Chen, and D. W. Cheung, "Graph topic scan statistic for spatial event detection," in *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management,* (2016), pp. 489–498.
13. N. Wu, F. Chen, J. Li, J. Huai, B. Zhou, N. Ramakrishnan *et al.*, "A nonparametric approach to uncovering connected anomalies by tree shaped priors," IEEE Transactions on Knowl. Data Eng. **31**, 1849–1862 (2018).
14. C. C. Aggarwal and K. Subbian, "Event detection in social streams," in *Proceedings of the 2012 SIAM international conference on data mining,* (SIAM, 2012), pp. 624–635.
15. P. Rozenshtein, F. Bonchi, A. Gionis, M. Sozio, and N. Tatti, "Finding events in temporal networks: Segmentation meets densest-subgraph discovery," in *2018 IEEE International Conference on Data Mining,* (2018).
16. M. Mongiovi, P. Bogdanov, R. Ranca, E. E. Papalexakis, C. Faloutsos, and A. K. Singh, "Netspot: Spotting significant anomalous regions on dynamic networks," in *Proceedings of the 2013 SIAM international conference on data mining,* (SIAM, 2013), pp. 28–36.
17. D. S. Johnson, M. Minkoff, and S. Phillips, "The prize collecting Steiner tree problem: theory and practice," in *Eleventh Annual ACM-SIAM Symposium on Discrete Algorithms,* (2000), pp. 760–769.