Supplementary Note: Why Banded Matrices Do Not Provide a Viable Exact Backend for Tree-Structured Semi-Markov CRF Inference

Contents

This note analyzes a natural optimization for exact semi-Markov CRF (semi-CRF) inference: using **banded / block-sparse matrix kernels** inside a **binary-tree (parallel-scan) formulation** of the dynamic program. The conclusion is negative: within this formulation, banded storage can reduce cost only at the smallest spans, but **cannot prevent near-dense intermediate operators** at the levels that dominate runtime and memory.

Scope. The claims below apply to *exact* tree-structured implementations that represent node-to-node combination as semiring matrix products over an expanded state space of size $\Theta(KC)$. This does not rule out approximate inference (e.g., pruning) or alternative exact formulations that do not materialize these operators.

Executive summary

A bounded maximum segment length K suggests "locality," so it is tempting to store intermediate operators in a banded format. Two structural facts defeat this approach:

1. **Local geometry mismatch.** At internal tree nodes, feasible combinations of left/right partial durations satisfy a **sum constraint** $d_1 + d_2 \leq S$ (where S is the node span). The induced sparsity over (d_1, d_2) is **triangular**, not diagonal-banded. Moreover, for moderate S, this pattern cannot be permuted into a *narrow* diagonal band (Proposition 1).

2. **Fill-in under composition.** Even for a genuinely banded "one-step" adjacency operator, repeated composition rapidly expands the support: in a boolean reachability model, the effective bandwidth of

m-step reachability grows as $\min(T, mK)$ (Lemma 2). In a balanced binary tree, m doubles per level, so the support becomes near-dense after $O(\log(T/K))$ levels.

Empirically, permutations (identity, "snake," Reverse Cuthill–McKee) reduce bandwidth only at very small spans; for $S \gtrsim K$ the best achievable bandwidth is close to dense width, matching the theory.

Notation

We consider a sequence of length T, equivalently boundary indices $0, 1, \ldots, T$. The maximum segment length is K, and the label/state count is C.

Two matrices are used as simplifying models:

Boundary reachability (global boolean model). Define $B \in \{0,1\}^{(T+1)\times(T+1)}$ by

$$B[i,j] = \begin{cases} 1 & \text{if } 1 \le j - i \le K \\ 0 & \text{otherwise} \end{cases}$$

This captures the monotone-time constraint "a single segment advances forward by at most K boundaries." For instance, with $T = 5$ and $K = 2$, the matrix B has $B[0,1] = B[0,2] = B[1,2] = B[1,3] = \cdots = 1$, but $B[0,3] = 0$ (since $3 - 0 = 3 > K$).

Duration compatibility at a tree node (local pattern). At a tree node of span length S, partial states are indexed by a duration $d \in \{1, \ldots, D\}$ where $D = \min(K, S)$, and a label $y \in \{1, \ldots, C\}$. The quantity D caps the duration states at the actual feasible range for the current span: durations cannot exceed the span S, nor can they exceed the global maximum K. Feasible left/right duration pairs satisfy $d_1 + d_2 \le S$. For each feasible duration pair, label interactions are typically dense, yielding *block-dense* sparsity.

Effective bandwidth

For a square matrix M whose indices have been flattened to $\{1, \ldots, n\}$, define the (structural) bandwidth

$$\mathrm{bw}(M) \;=\; \max\{\, |i - j| \;:\; M[i,j] \text{ is structurally nonzero} \,\}.$$

For boolean matrices, "nonzero" means 1; for semiring score matrices, it means "not masked / not $-\infty$."

A banded representation with half-bandwidth b requires $\mathrm{bw}(M) \le b$ (after any chosen permutation).

1. Motivation: why a banded backend seems plausible

Bounded segment length implies one-step locality: from boundary i, a single segment can reach only $\{i+1, \ldots, i+K\}$, so B is banded with bandwidth K. In a tree-structured implementation, one might hope that intermediate operators remain "approximately banded," yielding memory reductions from $O((KC)^2)$ toward $O(K^2C)$ or similar.

The key issue is that exact inference must represent *all* ways to compose segments, and composition expands support much faster than one-step locality would suggest.

2. Local incompatibility pattern is triangular, not diagonal-banded

At an internal tree node of span S, a standard combine operation must account for left and right partial durations d_1, d_2 that together fit inside the span:

$$d_1 + d_2 \le S.$$

In the (d_1, d_2) plane, this feasible set is an anti-diagonal triangle. A diagonal band, by contrast, corresponds to $|d_1 - d_2| \le b$. These shapes overlap only partially, and the mismatch becomes pronounced as soon as S is a nontrivial fraction of K.

To make "not narrow-banded under any ordering" precise, it suffices to lower bound the minimum achievable bandwidth.

Proposition 1 (Bandwidth lower bound from a clique)

Fix S, let $D = \min(K, S)$, and define a binary matrix $M_S \in \{0,1\}^{(DC)\times(DC)}$ indexed by pairs $(d, y) \in \{1, \dots, D\} \times \{1, \dots, C\}$, with

$$M_S[(d_1, y_1), (d_2, y_2)] = 1 \quad \Longleftrightarrow \quad d_1 + d_2 \le S.$$

Proposition 1. For any simultaneous row/column permutation π,

$$\mathrm{bw}\big(P_\pi^\top M_S P_\pi\big) \;\ge\; C\Big\lfloor \frac{S}{2} \Big\rfloor - 1.$$

Proof. Let $m = \lfloor S/2 \rfloor$. For any $d, d' \le m$, $d + d' \le 2m \le S$, so all states in

$$U = \{1, \dots, m\} \times \{1, \dots, C\}$$

are mutually adjacent in the undirected graph induced by M_S; hence U is a clique of size $|U| = mC$. For any linear ordering of a clique on n vertices, the first and last vertices are adjacent, so the bandwidth is at least $n - 1$. □

Consequence. When S is moderate (e.g., $S \approx K$) and C is not tiny, the best achievable bandwidth is already a large fraction of the dense width DC. In the extreme case $S \ge 2K$—since $D = K$ when $S \ge K$, the condition $d_1 + d_2 \le S$ with $d_1, d_2 \le K$ is always satisfied when $S \ge 2K$—the matrix M_S is structurally dense.

3. Fill-in under composition in a boolean reachability model

Even if the local operator were genuinely banded, repeated composition widens the support.

Lemma 2 (Boolean powers widen linearly in the number of composed segments)

Let B be the boundary adjacency matrix defined above, and interpret multiplication over the boolean semiring (OR as addition, AND as multiplication). Then $(B^m)[i, j] = 1$ if and only if there exists a path of **exactly** m segments from i to j.

Lemma 2. For $m \ge 1$,

$$(B^m)[i, j] = 1 \quad \Longleftrightarrow \quad m \le j - i \le mK,$$

and therefore

$$\mathrm{bw}(B^m) = \min(T,\, mK).$$

Proof. A path of m segments advances by $\delta_1 + \cdots + \delta_m$ with each $\delta_t \in \{1, \dots, K\}$, hence $m \le j - i \le mK$. Conversely, if $r = j - i$ satisfies $m \le r \le mK$, then r can be written as a sum of m integers in $[1, K]$ (e.g., start from all ones and distribute the remaining $r - m$ units without exceeding K). □

Implication for a balanced binary tree

In a balanced binary tree, combining two children corresponds (in this simplified reachability model) to composing reachability over a larger number of segments. The relevant parameter m therefore grows geometrically with tree level, so $\mathrm{bw}(B^m)$ increases rapidly and saturates at the dense width on the order of $\log_2(T/K)$ levels.

This is precisely what is observed in the illustrative plots: the reachability pattern becomes near-dense well before the root, so a banded container offers little benefit at the levels that dominate the total work.

4. Why matrix reordering helps only at very small spans

A reasonable countermeasure is to apply bandwidth-reducing permutations (e.g., Reverse Cuthill–McKee). Proposition 1 already limits what is possible at a single node: cliques force large bandwidth regardless of ordering.

At the global scale, once composition produces near-complete reachability, permutations cannot help substantially. One clean sufficient condition is "undirected completeness": if for every unordered pair of indices $\{u, v\}$ at least one of $M[u, v]$ or $M[v, u]$ is structurally nonzero, then the induced undirected graph is a clique and the bandwidth is $n - 1$ under any ordering.

Empirical bandwidth ratios

The accompanying bandwidth report evaluates several orderings across (K, C, S). Summarizing $\mathrm{bw}_{\mathrm{best}}/(n-1)$ by span fraction S/K:

- **Small spans** ($S \leq K/2$): best-case bandwidth ratios ranged from ≈ 0.23 to ≈ 0.68 (mean ≈ 0.44).
- **Moderate spans** ($K/2 < S \leq K$): ratios ranged from ≈ 0.78 to ≈ 0.99 (mean ≈ 0.90).
- **Large spans** ($S \geq K$): ratios were ≥ 0.97 and often exactly 1.0.

Thus, reordering can reduce bandwidth when spans are very small, but offers negligible benefit once spans approach K, which is the regime that dominates higher tree levels.

5. Practical implications for exact semi-Markov backends

5.1 Limited payoff of banded formats in a tree backend

Even if banded kernels are beneficial at the lowest tree levels (where spans are small and sparsity is substantial), the computation and memory footprint of tree-structured backends are dominated by higher levels, where:

- local duration compatibility becomes close to dense (Proposition 1), and
- multi-step composition produces near-dense support (Lemma 2).

Consequently, a tree backend must still materialize large near-dense intermediates, so banded storage does not change the practical memory cliff.

5.2 Block-triangular formats: geometrically matched, but limited on GPUs (empirical)

A geometrically better match than diagonal bandedness is a **block-triangular** format that stores only (d_1, d_2) blocks with $d_1 + d_2 \leq S$, each block being dense $C \times C$. This reduces storage by about a factor of two at the relevant spans.

In our PyTorch/CUDA experiments, however, this did not translate into a speedup at typical sparsity levels: the overhead of indirect indexing and non-contiguous memory access outweighed the savings from skipping a modest fraction of blocks. This observation is implementation- and hardware-dependent, but it is consistent with a common GPU performance pattern: moderate block sparsity often underperforms well-tuned dense kernels unless sparsity is high and the sparsity pattern is regular.

5.3 What remains viable

For *exact* inference in memory-limited regimes, the practical recommendation is therefore the same as in the main paper:

- use a **dense, vectorized linear scan** (with streamed potentials), which has predictable $O(TKC^2)$ work and $O(KC^2)$ working memory (including provided semirings), or
- use the PyTorch/Triton **streaming semi-CRF with checkpointing**, which achieves the same $O(TKC^2)$ work with $O(BKC)$ working memory via ring buffers and periodic checkpoints.

Both approaches are *exact*: they compute the correct partition function and gradients. The Triton streaming kernel may exhibit minor floating-point non-determinism due to warp-level atomic reductions, but this affects only bitwise reproducibility across runs—not the mathematical correctness of the result.

The key shared property is that neither approach materializes the $O((KC)^2)$ intermediate operators required by tree-structured backends. Instead, they process the sequence sequentially (or in checkpointed segments), avoiding the memory cliff that makes tree backends impractical for large K or C.

If sparsity is essential, it must be introduced by **approximation** (pruning / filtering), which enforces value sparsity rather than relying on structural sparsity that does not persist under composition. Recent work has explored segment filtering approaches that prune unlikely segments using lightweight classifiers before semi-CRF inference (Zaratiana et al., 2023), or hybrid architectures that combine token-level and segment-level models (Ye & Ling, 2018; Kong et al., 2016).

References

- G. E. Blelloch (1990). Prefix sums and their applications. Technical Report CMU-CS-90-190, Carnegie Mellon University.
- E. Cuthill and J. McKee (1969). Reducing the bandwidth of sparse symmetric matrices. *Proceedings of the 24th National Conference of the ACM*, 157–172.
- A. George and J. W. H. Liu (1981). *Computer Solution of Large Sparse Positive Definite Systems.* Prentice-Hall.
- L. Kong, C. Dyer, and N. A. Smith (2016). Segmental Recurrent Neural Networks. *ICLR*.
- A. M. Rush (2020). Torch-Struct: Deep Structured Prediction Library. *ACL*.
- S. Sarawagi and W. W. Cohen (2004). Semi-Markov Conditional Random Fields for Information Extraction. *NeurIPS*.
- Z. Ye and Z.-H. Ling (2018). Hybrid semi-Markov CRF for neural sequence labeling. *Proceedings of ACL*, 235–240.
- U. Zaratiana, N. Tomeh, N. El Khbir, P. Holat, and T. Charnois (2023). Filtered Semi-Markov CRF. *Findings of EMNLP*, 222–235.