EXPRESSIVITY LIMITS AND EXACTLY SOLVABLE REGIMES OF DEEP SUBMODULAR FUNCTIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Deep Submodular Functions (DSFs) compose modular maps and concave activations through nonnegative layers. Depth strictly enlarges the class, but no depth reaches all submodular functions, and even shallow DSFs can be hard to maximize. We show that composition rules impose all-depth expressivity constraints, while support organization provides structural certificates for exact cardinality-constrained maximization. On the expressivity side, we prove that DSF extensions built from activations that are C^2 on $(0, \infty)$ have completely positive negative Hessians at every interior point, regardless of depth. This curvature restriction is stronger than concavity and lattice-submodularity and yields smooth functions outside this extension class. A curvature support graph admits such a non-CP witness if and only if it contains an odd cycle of length at least five. For set functions, a clique-overcount inequality recovers the known exclusion of the graphic matroid rank of K_4 and lifts it to every matroid with an $M(K_4)$ minor. On the optimization side, bounded effective treewidth permits exact junction-tree dynamic programming at any depth, and its single-exponential dependence on width is necessary under the Exponential Time Hypothesis. Computation trees with disjoint child supports and no external modular tail form a second, incomparable exact regime that stays tractable at maximal effective width. Curvature obstruction and optimization hardness obey different graph conditions. Finally, we overcome the discrete expressivity obstruction by allowing controlled sums and inf-convolution of shallow atoms. This enlarged composition class represents every normalized monotone submodular function, with submodular intermediates and possibly exponential size. The expressivity gap therefore lies in the composition rules, not in the components. Feature-selection objectives learned from data instantiate both exact regimes, and taxonomy-based data selection on CIFAR-100 complements them.

1 INTRODUCTION

Monotone submodular functions model diminishing returns in machine learning, including training-data selection (Killamsetty et al., 2021; Mirzasoleiman et al., 2020), document summarization (Lin & Bilmes, 2011), sensor placement (Krause & Guestrin, 2005), and active learning (Wei et al., 2015). Their practical value depends on both the interactions they can represent and the cost of optimizing them. For cardinality-constrained maximization, greedy achieves the tight $(1 - 1/e)$ guarantee in the value-oracle model (Nemhauser et al., 1978; Nemhauser & Wolsey, 1978). Exact optimization therefore requires additional structure in the objective.

Deep Submodular Functions (DSFs) (Bilmes & Bai, 2017) encode such objectives through layers of nonnegative linear maps and concave nondecreasing activations. This construction guarantees submodularity and supports learning through a concave continuous extension (Dolhansky & Bilmes, 2016; Bai et al., 2018). Increasing depth expands the represented family without making it universal (Bilmes & Bai, 2017), while even depth-1 coverage objectives can be hard to optimize. Thus depth alone gives neither a certificate of representability nor a guarantee of tractability. For a learned DSF, this raises two concrete structural issues: which interactions its composition rules exclude, and which dependencies an exact optimizer must reconcile.

Composition rules impose all-depth expressivity constraints, while support organization provides certificates for exact cardinality-constrained maximization. Effective supports record ground elements

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.

that can influence each nonlinear component. Small separators let an optimizer reconcile overlapping choices, whereas disjoint computation branches permit independent budget allocation.

First, we establish all-depth expressivity obstructions. For DSF extensions built from activations that are C^2 on $(0, \infty)$, the negative Hessian is completely positive at every interior point, excluding some smooth monotone concave lattice-submodular functions. The classical characterization of completely positive graphs (Kogan & Berman, 1993) identifies precisely which curvature support graphs admit a non-CP witness. For cube traces on the edges of K_4, pairwise additivity forces the sum of triangle-level subadditivity deficits to be at most the global deficit. This quantitative inequality explains the known exclusion of the graphic matroid rank of K_4 (Bilmes & Bai, 2017), and closure under restriction and normalized contraction extends it to every matroid with an $M(K_4)$ minor.

Second, we identify two incomparable routes to exact cardinality-constrained optimization. At any depth, bounded effective treewidth permits top-layer factors to be tabulated and optimized by standard junction-tree DP. Under the Exponential Time Hypothesis (ETH), subexponential dependence on width is impossible, even for unit-cap pairwise factors. This worst-case bound coexists with exactly solvable computation trees of effective width $n - 1$: disjoint child supports and no external modular tail permit independent budget allocation through concave composition. In this disjoint regime, ordinary marginal-gain greedy is also exact; budget merging exploits the same structure to compute the optimal budget curve. These optimization certificates differ from the curvature criterion. Five-cycle graphs admit non-CP curvature witnesses but give bounded-width DSF optimization, whereas bipartite graphs forbid such witnesses yet support NP-hard DSF maximization.

Finally, we recover universality by enlarging the composition rules. A deletion–contraction construction assembles every normalized monotone submodular function from rank-one depth-1 SCMM atoms through nonnegative sums and controlled inf-convolution. Every intermediate output is submodular, although the expression may be exponential in size. The separation and construction locate the discrete expressivity gap in the composition rules. On HAR and ISOLET, we learn objectives in both exact regimes and compare them with modular, dense pairwise, and randomized-structure baselines. CIFAR-100 data selection illustrates the disjoint regime on a larger ground set.

2 PRELIMINARIES

2.1 SUBMODULAR FUNCTIONS AND DSF REPRESENTATIONS

A set function $f : 2^V \to \mathbb{R}$ on a finite ground set V ($|V| = n$) is *submodular* if $f(A) + f(B) \geq f(A \cup B) + f(A \cap B)$ for all $A, B \subseteq V$, *monotone* if $A \subseteq B \Rightarrow f(A) \leq f(B)$, and *normalized* if $f(\emptyset) = 0$. We write $\mathrm{MonSub}(V)$ for the class of monotone submodular normalized functions on V (equivalently, polymatroids).

Definition 2.1 (Deep Submodular Function)**.** A *depth-K DSF* on ground set V is defined by a layered computation network with input layer $U_0 = V$, nonlinear layers $U_1, \ldots, U_K$, and a nonnegative linear output, with edges directed from inputs toward higher layers. For each input $i \in V$, set $z_i(x) = x_i$. For each nonlinear node $u \in U_\ell$ ($\ell \geq 1$), define

$$z_u(x) \;=\; \varphi_u\left(\sum_{v \in U_{\ell-1}} w_{uv}\, z_v(x)\right), \qquad w_{uv} \geq 0, \tag{1}$$

where $\varphi_u : \mathbb{R}_+ \to \mathbb{R}_+$ is concave, non-decreasing, and satisfies $\varphi_u(0) = 0$. For the gradient and Hessian results only (Sections 3.1 and 3.2), we additionally assume $\varphi_u \in C^2((0, \infty))$. The discrete expressivity results and both dynamic programs require no differentiability and allow capped activations such as $\varphi(t) = \min(t, c)$, $c \geq 0$. For a layer-1 node $u \in U_1$, write $m_u = (w_{ui})_{i \in V} \in \mathbb{R}_+^V$, so that $z_u(x) = \varphi_u(\langle m_u, x \rangle)$. The associated extension is

$$\tilde{F}(x) \;=\; \sum_{u \in U_K} a_u\, z_u(x) \;+\; \langle m_+, x \rangle, \qquad a_u \geq 0, \quad m_+ \in \mathbb{R}_+^V, \tag{2}$$

and the *DSF set function* is $F(A) := \tilde{F}(\mathbf{1}_A)$. For continuous-extension results, we additionally assume that the activations are continuous at zero. The discrete expressivity and optimization results require no boundary continuity.

Nonnegative weights, activations, output coefficients, and modular tail ensure $F \in \mathrm{MonSub}(V)$. We use the layered form in the main text. The equivalent DAG form is stated in Section A.

We write $\mathrm{DSF}_K(V)$ for the class of depth-K DSF set functions on V, and $\mathrm{DSF}(V) := \bigcup_{K\geq 1} \mathrm{DSF}_K(V)$. A depth-1 DSF is a sum of concave-over-modular (SCMM) functions: $F(A) = \sum_j \varphi_j(\langle m_j, \mathbf{1}_A\rangle) + \langle m_+, \mathbf{1}_A\rangle$, where each $m_j \in \mathbb{R}_+^V$ is the modular weight vector of layer-1 node j.

Definition 2.2 (Effective support and effective clique hypergraph)**.** For a layer-1 node u, its *effective support* is $E(u) = \mathrm{supp}(m_u)$. At higher layers, define

$$E(u) = \bigcup_{v\in \mathrm{pred}(u):\, w_{uv}>0} E(v),$$

where $\mathrm{pred}(u)$ denotes its predecessors. The *effective clique hypergraph* is $\mathcal{H}_F^{\mathrm{eff}} = (V, \{E(u) : u \text{ is a nonlinear node}\})$.

2.2 Continuous extensions and cube traces

The continuous extension $\tilde{F} : \mathbb{R}_+^V \to \mathbb{R}_+$ (Equation (2)) is concave on $\mathbb{R}_+^V$, hence in particular on $[0,1]^V$, and satisfies $\tilde{F}(\mathbf{1}_A) = F(A)$. When the activations are C^2 on $(0,\infty)$, $\tilde{F}$ is C^2 on the interior $\mathbb{R}_{++}^V$ (Section B.1). We call $F(A) = \tilde{F}(\mathbf{1}_A)$ the *cube trace* of $\tilde{F}$.

Continuous lattice-submodularity. For a C^2 function $g : \mathbb{R}_+^n \to \mathbb{R}$, we say g is *continuously lattice-submodular* if $\frac{\partial^2 g}{\partial x_i \partial x_j}(x) \leq 0$ for all $i \neq j$ throughout the domain. On product domains, this is the differential characterization of lattice-submodularity.

Completely positive and doubly nonnegative cones. For symmetric $n \times n$ matrices, let

$$\mathcal{CP}_n := \mathrm{cone}\{vv^\top : v \in \mathbb{R}_+^n\}, \qquad \mathcal{DNN}_n := \{M \succeq 0 : M_{ij} \geq 0\}.$$

We use the classical strict inclusion $\mathcal{CP}_n \subsetneq \mathcal{DNN}_n$ for $n \geq 5$ (Maxfield & Minc, 1962; Diananda, 1962).

3 Expressivity Gaps

Nonnegative composition imposes two all-depth restrictions: a curvature invariant for smooth extensions and a cross-set redundancy inequality for cube traces. The second is needed because excluding one continuous extension does not exclude all extensions of its vertex values.

3.1 Completely positive Hessians of DSF extensions

Nonnegative gradients and concave activations constrain the second-order chain rule at every layer. The resulting curvature invariant is independent of depth. The related gradient factorization is given in Section B.

Lemma 3.1 (DSF Hessian is completely positive)**.** *For a DSF extension $\tilde{F}$ with activations continuous at zero and C^2 on $(0,\infty)$, at every $x \in \mathbb{R}_{++}^V$,*

$$-\nabla^2 \tilde{F}(x) \;\in\; \mathcal{CP}_n.$$

Induction over the computation DAG shows that at each concave composition the negative Hessian decomposes into a completely positive rank-one term from the nonnegative gradient and a nonnegative weighted sum of lower-level completely positive Hessians. Details are in Section B.

Complete positivity requires the curvature to decompose along nonnegative directions. The next result shows concavity and continuous lattice-submodularity alone do not impose this restriction.

3.2 A CONTINUOUS CP-VS-DNN SEPARATION

Concavity and lattice-submodularity require a doubly nonnegative negative Hessian, which need not be completely positive. A concrete witness is the C_5-supported circulant matrix

$$H = \begin{pmatrix} 5 & 3 & 0 & 0 & 3 \\ 3 & 5 & 3 & 0 & 0 \\ 0 & 3 & 5 & 3 & 0 \\ 0 & 0 & 3 & 5 & 3 \\ 3 & 0 & 0 & 3 & 5 \end{pmatrix}, \qquad \begin{aligned} \lambda_{\min}(H) &= (7 - 3\sqrt{5})/2 > 0, \\ \operatorname{tr} H &= 25 < 30 = 2 \sum_{ij \in E(C_5)} H_{ij}. \end{aligned} \tag{3}$$

Every C_5-supported CP matrix satisfies $\operatorname{tr} M \geq 2 \sum_{ij \in E(C_5)} M_{ij}$, so $H \in \mathcal{DNN}_5 \setminus \mathcal{CP}_5$. The next theorem realizes any such matrix as curvature of a smooth monotone function. Verification of the witness and the construction are in Section D.

Theorem 3.2 (Continuous-extension expressivity gap)**.** *For every $n \geq 5$ and every $B \in \mathcal{DNN}_n \setminus \mathcal{CP}_n$, there exists a C^∞ function $F : \mathbb{R}^n_+ \to \mathbb{R}_+$ with $F(\mathbf{0}) = 0$ that is monotone non-decreasing, concave, and continuously lattice-submodular on $\mathbb{R}^n_+$, and satisfies*

$$-\nabla^2 F(x_0) = B$$

at some interior point $x_0 \in (0, \infty)^n$. In particular, F is not a DSF continuous extension with activations that are C^2 on $(0, \infty)$.

The obstruction also has a support characterization. For a graph G on V, say that a matrix is *G-supported* if its off-diagonal entries vanish outside $E(G)$. Kogan–Berman's theorem states that every G-supported doubly nonnegative matrix is completely positive exactly when G contains no odd cycle of length at least five (Kogan & Berman, 1993).

Corollary 3.3 (Support-sensitive curvature obstruction)**.** *If G contains an odd cycle of length at least five, there exists a smooth monotone concave lattice-submodular function whose negative Hessian is G-supported throughout $\mathbb{R}^V_+$ and is non-CP at an interior point. It is therefore not a DSF extension with $C^2((0, \infty))$ activations. If G contains no such cycle, every G-supported negative Hessian of a smooth concave lattice-submodular function is CP.*

The construction preserves off-diagonal zeros globally (Section D.4). Absence of this obstruction does not imply DSF representability. Non-representability of a set function requires excluding all DSF extensions with that cube trace. We therefore develop a separate obstruction directly in terms of set-function values.

3.3 A DISCRETE ALL-DEPTH SEPARATION VIA K_4 CLIQUE-OVERCOUNT

At the cube-trace level, the obstruction becomes combinatorial. A clique-overcount inequality on K_4 gives an all-depth exclusion, witnessed by the graphic matroid rank function on the six edges.

Let $V = E(K_4)$. The graph K_4 has four triangles $\mathcal{T} = \{T_1, T_2, T_3, T_4\}$, each a three-element subset of V. We use the following deficit notation to express the obstruction. The quantity ρ_f is the surplus of Bilmes & Bai (2017, Definition 5.19).

Definition 3.4 (Subadditivity deficit and Property P)**.** For a normalized set function $f : 2^V \to \mathbb{R}_+$, define the *subadditivity deficit*

$$\rho_f(S) \;:=\; \sum_{e \in S} f(\{e\}) - f(S).$$

We say f satisfies *Property* P if

$$\rho_f(\{e, e'\}) = 0 \;\text{ for all } e \neq e' \quad \Longrightarrow \quad \rho_f(V) \;\geq\; \sum_{T \in \mathcal{T}} \rho_f(T).$$

This is the clique-overcount inequality: if there is no pairwise redundancy, then triangle-level redundancy cannot exceed the redundancy available globally.

Theorem 3.5 (K_4 clique-overcount inequality)**.** *Every DSF cube trace on $V = E(K_4)$ satisfies Property* P*. The graphic matroid rank function $r_{M(K_4)}$ is pair-additive ($r(\{e, e'\}) = 2$ for all $e \neq e'$) but satisfies*

$$\rho_r(V) = 3 < 4 = \sum_{T \in \mathcal{T}} \rho_r(T),$$

hence violates P*. Therefore $r_{M(K_4)} \notin \mathrm{DSF}(V)$.*

The inequality explains the known exclusion of $r_{M(K_4)}$ (Bilmes & Bai, 2017) through a quantitative constraint across overlapping triangles. The surplus identities and qualitative preservation properties are established in Bilmes & Bai (2017, Section 5.4). Our proof inducts over the DSF: modular functions have zero deficit and nonnegative sums preserve Property P termwise. For the nontrivial concave-composition step, pair-additivity forces an initial linear segment of the activation. Its deficit from that linear part is a nonnegative mixture of one-sided hinges. The four-triangle hinge bound controls each contribution, preserving the cross-set inequality at every depth (Sections F and F.3).

Corollary 3.6 (Minor-lifted rank obstruction)**.** *If a matroid M has $M(K_4)$ as a minor, then its rank function is not a DSF cube trace of any depth.*

Restriction and normalized contraction preserve DSF representability. Consequently, the six-element witness certifies non-representability for every matroid containing it as a minor (Section F.4).

4 Exactly Solvable Regimes

The preceding invariants constrain what a single DSF can represent. We now examine when the support structure of a supplied representation permits exact cardinality-constrained maximization. Two incomparable structures enable it: small separators and disjoint computation branches.

For the decomposition, let $G_{\mathcal{H}}$ be the *primal graph* of $\mathcal{H}_F^{\mathrm{eff}}$, with vertex set V and an edge between distinct elements whenever they share an effective support. A *junction tree* is a tree of bags covering V and every hyperedge, with the bags containing any element forming a connected subtree. The width of a junction tree is its largest bag size minus one, and $\mathrm{tw}(\mathcal{H}_F^{\mathrm{eff}}) := \mathrm{tw}(G_{\mathcal{H}})$ is the minimum width over such decompositions (Lauritzen & Spiegelhalter, 1988). Width bounds refer to this ground-set graph. Figure 1 illustrates the effective clique hypergraph (Definition 2.2). Its ground-set scopes define local factors for junction-tree DP and disjoint child subproblems for computation-tree DP.

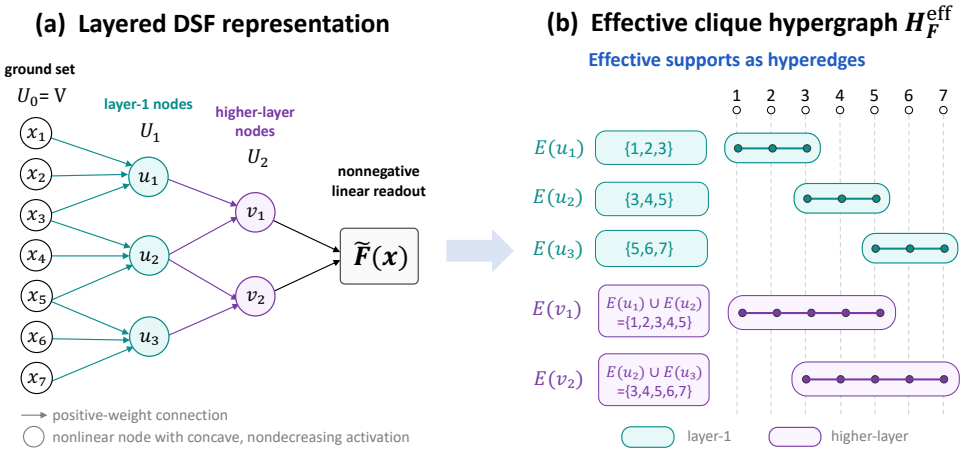


Figure 1: Effective clique hypergraph of a depth-2 DSF. Positive-weight paths in (a) induce the ground-set scopes in (b). Each nonlinear node contributes a hyperedge. The final linear output contributes no additional hyperedge.

We optimize a supplied representation under an integer budget $1 \leq k \leq |V|$ ($k = 0$ is immediate). Costs count unit-time arithmetic, comparisons, and activation evaluations. JT-DP assumes precomputed bag potentials, with preprocessing accounted for in Section H.

4.1 JUNCTION-TREE DP OVER EFFECTIVE FACTOR HYPERGRAPHS

For each top-layer node $u \in U_K$, define $f_u(S) = a_u z_u(\mathbf{1}_S)$ for $S \subseteq E(u)$. Locality of effective supports gives

$$F(A) = \sum_{u \in U_K} f_u(A \cap E(u)) + \langle m_+, \mathbf{1}_A \rangle. \tag{4}$$

Root $\mathcal{T}$ at r. Let $S_t = B_t \cap B_{\mathrm{par}(t)}$, $S_r = \emptyset$, and $R_t = B_t \setminus S_t$. The potential Φ_t sums factors assigned to bag t. The state $V_t(a, b)$ fixes its bag assignment a and counts b selected elements private to its subtree, excluding S_t. Separator messages reconcile shared selections while max-plus convolution $\oplus$ allocates the budget. Algorithm 1 gives the resulting recurrence.

Algorithm 1 Junction-tree DP for tabulated DSF factors

Require: Factor tables from (4), decomposition $\mathcal{T}$, budget k.
1: Assign each factor to a bag containing its support and form Φ_t.
2: **for** bags t in postorder **do**
3: $\quad V_t(a, b) \leftarrow \Phi_t(a)$ if $b = |a \cap R_t|$, and $-\infty$ otherwise.
4: $\quad$ **for** children c of t **do**
5: $\quad\quad M_c(\sigma, b) \leftarrow \max_{a_c \subseteq B_c:\, a_c \cap S_c = \sigma} V_c(a_c, b)$.
6: $\quad\quad V_t(a, \cdot) \leftarrow V_t(a, \cdot) \oplus M_c(a \cap S_c, \cdot)$ for each $a \subseteq B_t$.
7: $\quad$ **end for**
8: **end for**
9: **return** $\max_{a \subseteq B_r,\, b \le k} V_r(a, b)$.

All budget tables are truncated at k. Bag enumeration and convolution give the factors 2^{w+1} and k^2 in the bound below. The full state invariant and proof appear in Section H.

Theorem 4.1 (Junction-tree DP under bounded effective treewidth)**.** *Let F be a DSF of any depth whose effective clique hypergraph (Definition 2.2) admits a tree decomposition $\mathcal{T}$ of width w. Given $\mathcal{T}$, with each top-layer factor in* (4) *and each unary tail factor assigned to a bag containing its support and aggregated bag potentials Φ_t precomputed, the maximum $\max_{|A| \le k} F(A)$ is computed exactly in $O(|\mathcal{T}|\, 2^{w+1} k^2)$ time.*

The bound applies at arbitrary depth, with efficiency governed by the support graph. A top-layer scope of size s forces $w \ge s - 1$, although small scopes alone do not ensure small width. Tabulation gives general local factors, which need not be single concave-over-modular ridges. Overlap and shared subcomputations are allowed. The locality proof and preprocessing costs appear in Section H.

Theorem 4.2 (Necessary exponential width dependence)**.** *Assuming ETH, exact cardinality-constrained DSF maximization has no $2^{o(w)}\,\mathrm{poly}(L)$ algorithm, even for depth-1 unit-cap factors on pairs. Here L is the binary input length and w is the effective treewidth. The lower bound also holds when a decomposition is supplied and w denotes its width.*

For a graph G, the objective $F_G(A) = \sum_{uv \in E(G)} \min(\mathbf{1}[u \in A] + \mathbf{1}[v \in A], 1)$ counts covered edges and has effective graph G. Its optimum equals $|E(G)|$ exactly when G has a vertex cover of size at most k. The standard sparse-SAT reduction proves Theorem 4.2 (Section I). It makes the single-exponential width dependence necessary in the worst case, without asserting optimality of the exponential base or the budget factor.

This hardness concerns worst-case families, not every large-width instance. The next regime remains exactly solvable even at maximal effective width.

4.2 DISJOINT COMPOSITION BEYOND BOUNDED WIDTH

A computation tree has no shared subcomputations and includes the linear output as an identity-activation root. Child effective supports are disjoint at every internal node, including this root, and there is no external modular tail. Let $V_u(b)$ be the optimum of node u's subfunction with at most b

elements from $E(u)$. At internal nodes, disjointness gives

$$V_u(b) = \varphi_u \left(\max_{\substack{\sum_{c \in \mathrm{ch}(u)} b_c = b \\ b_c \in \mathbb{Z}_{\geq 0}}} \sum_{c \in \mathrm{ch}(u)} w_{uc} V_c(b_c) \right). \tag{5}$$

Child optima combine independently, and monotonicity permits maximizing the weighted sum before applying φ_u. Leaf tables use sorted weights.

Corollary 4.3 (Exact optimization on disjoint computation trees)**.** *For a DSF computation tree with pairwise disjoint child effective supports and no external modular tail, the cardinality-constrained maximum is computed in $O(P_{\mathrm{leaf}} + |\mathcal{L}|dk)$ time. Here $|\mathcal{L}|$ counts tree nodes, $d \geq 1$ bounds arity, and $P_{\mathrm{leaf}} = \sum_\ell |E(\ell)| \log(1 + |E(\ell)|)$ is the leaf-weight sorting cost. No bound on effective treewidth is required.*

These budget tables are non-decreasing and discretely concave (Lemma K.2). Each merge therefore takes $O(k)$ time by merging sorted child marginals, the classical slope-merge operation (de Berg et al., 2008). The proofs and aligned laminar-matroid extension appear in Sections J, J.1 and K.

Corollary 4.4 (Greedy exactness on disjoint computation trees)**.** *Under the hypotheses of Corollary 4.3, ordinary marginal-gain greedy, with arbitrary tie-breaking, returns an optimum of $\max_{|A| \leq k} F(A)$ for every cardinality budget k.*

The proof establishes optimality of every greedy prefix through disjoint sums and concave composition (Section K.3).

Example 4.5 (Exact optimization at maximal effective width)**.** For even $n \geq 4$, split $V = V_1 \sqcup V_2$ equally and set $F(A) = \sqrt{\sqrt{|A \cap V_1|} + \sqrt{|A \cap V_2|}}$. Its disjoint computation tree permits exact optimization, yet every DSF representation has effective width $n - 1$: all pairwise mixed differences are nonzero, forcing every additive factorization to have complete primal graph (Section N.1).

Conversely, the width-three DSF in Proposition K.4 has optimal-value budget table $(0, 7, 10, 14, 14)$. Its marginal values $(7, 3, 4, 0)$ are not non-increasing. By Lemma K.2, it admits no disjoint computation-tree representation without a modular tail. The two regimes are therefore incomparable at the function level.

4.3 Distinct expressivity and optimization boundaries

Beyond exactness, depth-1 coverage DSFs retain the $(1 - 1/e + \varepsilon)$ approximation hardness of Max-k-Cover for every fixed $\varepsilon > 0$, unless $\mathrm{P} = \mathrm{NP}$ (Theorem I.1). For fixed minor-closed support families, bounded treewidth gives polynomial-time exact optimization, whereas unbounded treewidth permits hard unit-cap instances (Corollary I.2). Both concern worst-case families. The disjoint regime shows large width alone does not rule out exact optimization of a particular objective.

The support conditions behind curvature separation and exact optimization differ. For a smooth DSF, its Hessian support at each point is contained in its effective primal graph. An excluded target instead has only a curvature support graph. The five-cycle permits a target excluded by the smooth-extension invariant (Corollary 3.3), while DSFs with effective graph C_5 have width two and admit exact JT-DP. On bipartite supports the CP test is silent, yet maximizing F_G is NP-hard by bipartite maximum vertex cover hardness (Joret & Vetta, 2015, Theorem 4.1). Thus these support families separate the curvature obstruction from worst-case optimization hardness.

5 Controlled Composition Beyond a Single DSF

The K_4 obstruction survives every depth of a single DSF. We now retain rank-one depth-1 SCMM atoms and allow nonnegative sums and controlled inf-convolution, recovering universality with submodular intermediate outputs.

Definition 5.1 (Compositional DSF class)**.** The class $\mathrm{CDSF}(V)$ consists of all functions $f \in \mathrm{MonSub}(V)$ that admit a finite expression built from elements of $\mathrm{DSF}(V)$ using:

(S1) finite nonnegative sums.

(S2) inf-convolution: $(f \square g)(A) := \min_{B \subseteq A}(f(B) + g(A \setminus B))$.

Every intermediate output is required to belong to $\mathrm{MonSub}(V)$.

Inf-convolution does not generally preserve submodularity. The deletion–contraction construction below verifies the requirement inductively: nonnegative sums and dummy lifts preserve MonSub, and each inf-convolution output is identified with its submodular target by Lemmas M.3 and M.4.

Theorem 5.2 (Controlled recursive construction)**.** *Every $f \in \mathrm{MonSub}(V)$ admits a finite recursive expression from rank-one depth-1 SCMM atoms using nonnegative sums and controlled inf-convolution steps. Each intermediate output remains in* MonSub*. The construction may have size exponential in $|V|$.*

To see the mechanism, fix $e \in V$, set $W = V \setminus \{e\}$ and $\alpha = f(\{e\})$, and recursively represent deletion $d(S) = f(S)$ and normalized contraction $c(S) = f(S \cup \{e\}) - \alpha$ on W. Define

$$G_e(A) = d(A \cap W) + \alpha \mathbf{1}[e \in A],$$
$$H_e(A) = c(A \cap W) + \alpha \mathbf{1}[A \neq \emptyset].$$

The added terms are rank-one atoms. Both envelopes are submodular, dominate f, and satisfy $\min(G_e, H_e) = f$. Subadditivity gives $G_e \square H_e = f$, so each recursive merge has a submodular output. This identifies the assembly operation that overcomes the discrete obstruction while retaining the original shallow components (Section M).

6 Experiments

We test exactness and computational cost on synthetic instances, then ask whether both tractable families yield useful objectives on real data. Protocols and full results are in Section O.

6.1 Numerical validation of exact optimization

On 960 depth-1 instances of width two or three, JT-DP agrees with exhaustive search within 7.1×10^{-15}. Greedy is suboptimal on 29.3%, with worst ratio 0.765, demonstrating the role of exact optimization (Section O.2). On 720 disjoint computation trees, generic and accelerated DP agree within 4.3×10^{-14}. Acceleration reaches $24.3\times$ at $(n, k, d) = (500, 100, 2)$ (Sections O.6 and O.7).

6.2 Feature selection with learned tractable objectives

A learned Chow–Liu tree combines relevance with pairwise redundancy penalties. A modular shift preserves its fixed-cardinality maximizers and yields a width-one DSF. For the disjoint regime, correlation clustering learns a partition $\mathcal{P}$ and defines

$$F_{\mathrm{cap}}(A) = \sum_{g \in \mathcal{P}} \min\left\{ \sum_{i \in A \cap g} r_i,\ \alpha \max_{i \in g} r_i \right\},$$

where $r_i \geq 0$ is feature relevance and $\alpha \geq 0$ controls the reward cap within each group. The learned groups need not have small supports, but their disjointness permits exact budget allocation. JT-DP and budget merging optimize the tree and group objectives, respectively. We compare them with modular relevance and a dense pairwise objective optimized by greedy, using size-matched random groups to test the value of learned grouping.

On HAR (561 features) and ISOLET (617 features), structures and parameters are selected by grouped training-only validation. A common linear classifier evaluates selected features on the official subject-disjoint and speaker-disjoint test splits (Section O.8).

Figure 2 compares test accuracy across the selection methods. On ISOLET, group caps improve substantially over modular selection and remain within 1.3 percentage points of dense pairwise selection. They outperform the mean of 20 independently tuned, size-matched random partitions at all three budgets (Section O.8). Together, these comparisons indicate that learned disjoint groups capture useful redundancy while retaining exact solvability. On HAR, group caps perform best at two budgets and the tree DSF at the remaining budget, showing that the useful support organization depends on the dataset and budget.

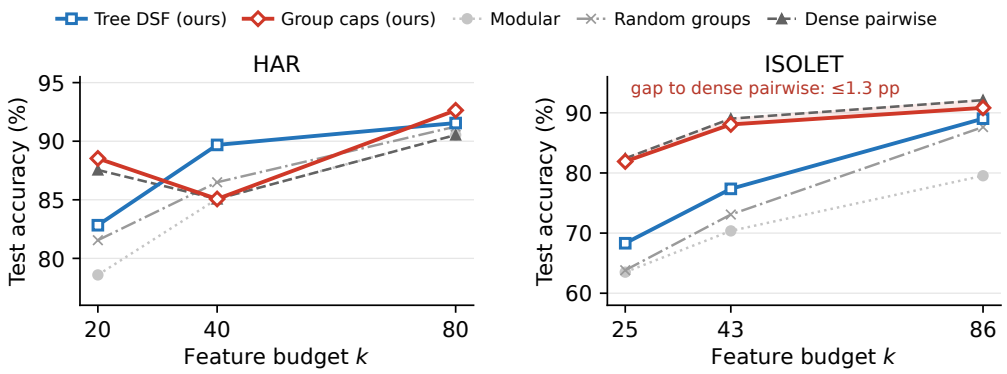


Figure 2: Feature-selection accuracy. Colored curves denote exactly solvable families; random groups average 20 size-matched partitions.

6.3 Data selection with taxonomy-based objectives

HC-caps uses frozen CIFAR-100 features and the official taxonomy. At $k = 500, 1000$, its upper caps are non-binding, giving a disjoint fine-class cap objective. DP matches lazy greedy's objective value by Corollary 4.4 and is respectively $24.4\times$ and $16.8\times$ faster (Table 6). Training ResNet-18 on selected subsets improves accuracy by 2.8–3.1 percentage points over uniform and balanced random selection. Against global, class-balanced, and taxonomy-matched facility location, accuracy differences range from -0.60 to $+0.28$ points, with paired 95% intervals containing zero. These results support selection at low computational cost for this objective (Sections O.9 and O.10).

7 Related Work

Submodular optimization and structured tractability. The $(1-1/e)$ value-oracle guarantee is tight under cardinality constraints (Nemhauser et al., 1978; Nemhauser & Wolsey, 1978). Continuous greedy attains the same ratio under general matroid constraints (Calinescu et al., 2011). Bai et al. (2018) use the DSF concave extension and pipage rounding for matroid-constrained approximation, with a guarantee that can exceed $1 - 1/e$ under their representation conditions. We obtain exact optimization under cardinality constraints for bounded-width DSFs and disjoint computation trees.

Deep and neural submodular function classes. DSFs strictly extend SCMMs but are not universal (Bilmes & Bai, 2017; Dolhansky & Bilmes, 2016). Related architectures include FlexSubNet's learned concave modules (De & Chakrabarti, 2022), EDSFs (Hosseini et al., 2024), and DSPNs (Bhatt et al., 2024). EDSFs represent all monotone set functions without guaranteeing submodularity. Bilmes & Bai (2017, Definition 5.19, Section 5.4, Theorem 6.26) introduced surplus and its preservation properties and proved that DSFs cannot represent the graphic matroid rank of K_4. We strengthen this exclusion with the quantitative all-depth inequality P and minor closure, and establish a complementary CP-Hessian obstruction for smooth extensions.

Structural tools: treewidth and complete positivity. Junction-tree inference (Lauritzen & Spiegelhalter, 1988) and sorted marginal merging (de Berg et al., 2008) supply the algorithmic primitives. Our results identify their DSF support conditions and establish width sharpness. The curvature separation uses the classical CP/DNN cone gap (Maxfield & Minc, 1962; Diananda, 1962; Berman & Shaked-Monderer, 2003) and CP graph characterization (Kogan & Berman, 1993).

8 Conclusion

DSF expressivity and exact optimization are governed by distinct structural mechanisms. Nonnegative composition imposes continuous and discrete restrictions that persist at arbitrary depth. Recovering universality through controlled inf-convolution of shallow atoms locates the expressive gap in the composition rules. For optimization, small separators and disjoint computation branches provide two routes to exactness, with the latter remaining tractable at maximal effective width. These findings support a structural approach to submodular modeling: composition rules specify the expressive class, while separability of effective supports provides verifiable conditions for exact optimization.

REPRODUCIBILITY STATEMENT

All theoretical claims are stated with their assumptions in the main text and proved in full in the appendix: the CP-Hessian characterization in Section B, the continuous separation in Section D, the minor-lifted rank obstruction in Section F.4, junction-tree correctness and complexity in Section H, and the width lower bound with its input model in Section I. The experimental protocols, including estimator settings, the grouped training-only validation used to select every structure and hyperparameter, and the seeds, are given in Section O. The experiments use the public UCI HAR, UCI ISOLET, and CIFAR-100 datasets with their official splits. The supplementary material provides the code for all objectives and solvers, the synthetic experiments, the feature-selection accuracy experiments, and the CIFAR-100 subset selection and its timing; the downstream classifiers follow the training recipe stated in Section O.9.

ETHICS STATEMENT

This work is a theoretical study of deep submodular functions, evaluated on three public benchmark datasets: UCI HAR, UCI ISOLET, and CIFAR-100. We conducted no experiments with human subjects, collected no new data, and release no new dataset. The HAR recordings were gathered by their original authors under their own consent procedures, and we use only the released, de-identified feature representation with its official subject-disjoint split.

AI USE STATEMENT

We used generative AI tools to refine the writing of this paper and to search for related work. We also used them to check proofs. We have reviewed all of this AI-assisted content ourselves. We take responsibility for the final content of this work, including all text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Wenruo Bai, William Stafford Noble, and Jeff A Bilmes. Submodular maximization via gradient ascent: The case of deep submodular functions. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/b43a6403c17870707ca3c44984a2da22-Paper.pdf.

Abraham Berman and Naomi Shaked-Monderer. *Completely Positive Matrices*. World Scientific, 2003. doi: 10.1142/5273.

Gantavya Bhatt, Arnav M Das, and Jeffrey A Bilmes. Deep submodular peripteral networks. *Advances in Neural Information Processing Systems*, 37:56552–56587, 2024.

Jeffrey Bilmes and Wenruo Bai. Deep submodular functions. *arXiv preprint arXiv:1701.08939*, 2017.

Hans L. Bodlaender. A linear time algorithm for finding tree-decompositions of small treewidth. In *Proceedings of the Twenty-Fifth Annual ACM Symposium on Theory of Computing*, STOC '93, pp. 226–234, New York, NY, USA, 1993. Association for Computing Machinery. ISBN 0897915917. doi: 10.1145/167088.167161. URL https://doi.org/10.1145/167088.167161.

Gruia Calinescu, Chandra Chekuri, Martin Pál, and Jan Vondrák. Maximizing a monotone submodular function subject to a matroid constraint. *SIAM J. Comput.*, 40(6):1740–1766, December 2011. ISSN 0097-5397. doi: 10.1137/080733991. URL https://doi.org/10.1137/080733991.

Ron Cole and Mark Fanty. Isolet. UCI Machine Learning Repository, 1994.

Abir De and Soumen Chakrabarti. Neural estimation of submodular functions with applications to differentiable subset selection. *Advances in Neural Information Processing Systems*, 35:19537–19552, 2022.

Mark de Berg, Otfried Cheong, Marc van Kreveld, and Mark Overmars. *Computational Geometry: Algorithms and Applications*. Springer, third edition, 2008. doi: 10.1007/978-3-540-77974-2.

P. H. Diananda. On non-negative forms in real variables some or all of which are non-negative. *Mathematical Proceedings of the Cambridge Philosophical Society*, 58(1):17–25, 1962. doi: 10.1017/S0305004100036185.

Brian W Dolhansky and Jeff A Bilmes. Deep submodular functions: Definitions and learning. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper_files/paper/2016/file/7fea637fd6d02b8f0adf6f7dc36aed93-Paper.pdf.

Uriel Feige. A threshold of ln n for approximating set cover. *J. ACM*, 45(4):634–652, July 1998. ISSN 0004-5411. doi: 10.1145/285055.285059. URL https://doi.org/10.1145/285055.285059.

M. R. Garey, David S. Johnson, and Larry J. Stockmeyer. Some simplified NP-complete graph problems. *Theor. Comput. Sci.*, 1:237–267, 1976. URL https://api.semanticscholar.org/CorpusID:205090369.

Seyed Mohammad Hosseini, Arash Jamshid, Seyed Mahdi Noormousavi, Mahdi Jafari Siavoshani, and Naeimeh Omidvar. Extended deep submodular functions. *arXiv preprint arXiv:2409.12053*, 2024.

R. Impagliazzo, R. Paturi, and F. Zane. Which problems have strongly exponential complexity? In *Proceedings 39th Annual Symposium on Foundations of Computer Science (Cat. No.98CB36280)*, pp. 653–662, 1998. doi: 10.1109/SFCS.1998.743516.

Gwenaël Joret and Adrian Vetta. Reducing the rank of a matroid. *Discrete Mathematics & Theoretical Computer Science*, 17(Discrete Algorithms), 2015.

Krishnateja Killamsetty, Durga Sivasubramanian, Ganesh Ramakrishnan, and Rishabh Iyer. Glister: Generalization based data subset selection for efficient and robust learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 8110–8118, 2021. doi: 10.1609/aaai.v35i9.16988.

Natalia Kogan and Abraham Berman. Characterization of completely positive graphs. *Discrete Mathematics*, 114(1):297–304, 1993. doi: 10.1016/0012-365X(93)90374-3.

Andreas Krause and Carlos Guestrin. Near-optimal nonmyopic value of information in graphical models. In *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence*, UAI'05, pp. 324–331, Arlington, Virginia, USA, 2005. AUAI Press. ISBN 0974903914.

S. L. Lauritzen and D. J. Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society: Series B (Methodological)*, 50(2):157–194, 01 1988. ISSN 0035-9246. doi: 10.1111/j.2517-6161.1988.tb01721.x. URL https://doi.org/10.1111/j.2517-6161.1988.tb01721.x.

Hui Lin and Jeff Bilmes. A class of submodular functions for document summarization. In Dekang Lin, Yuji Matsumoto, and Rada Mihalcea (eds.), *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 510–520, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL https://aclanthology.org/P11-1052/.

John E. Maxfield and Henryk Minc. On the matrix equation $X'X = A$. *Proceedings of the Edinburgh Mathematical Society*, 13(2):125–129, 1962. doi: 10.1017/S0013091500014681.

Baharan Mirzasoleiman, Jeff Bilmes, and Jure Leskovec. Coresets for data-efficient training of machine learning models. In *International Conference on Machine Learning*, pp. 6950–6960. PMLR, 2020.

G. L. Nemhauser and L. A. Wolsey. Best algorithms for approximating the maximum of a submodular set function. *Mathematics of Operations Research*, 3(3):177–188, 1978. ISSN 0364765X, 15265471. URL http://www.jstor.org/stable/3689488.

George L. Nemhauser, Laurence A. Wolsey, and Marshall L. Fisher. An analysis of approximations for maximizing submodular set functions—i. *Mathematical Programming*, 14:265–294, 1978. URL https://api.semanticscholar.org/CorpusID:206800425.

Jorge Reyes-Ortiz, Davide Anguita, Alessandro Ghio, Luca Oneto, and Xavier Parra. Human activity recognition using smartphones. UCI Machine Learning Repository, 2012.

Neil Robertson and P D Seymour. Graph minors. v. excluding a planar graph. *J. Comb. Theory Ser. B*, 41(1):92–114, August 1986. ISSN 0095-8956. doi: 10.1016/0095-8956(86)90030-4. URL https://doi.org/10.1016/0095-8956(86)90030-4.

R. Tyrrell Rockafellar. *Convex Analysis*. Princeton Landmarks in Mathematics and Physics. Princeton University Press, 1970. ISBN 978-1-4008-7317-3.

Kai Wei, Rishabh Iyer, and Jeff Bilmes. Submodularity in data subset selection and active learning. In Francis Bach and David Blei (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 1954–1963, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/wei15.html.

A Additional Notation and Preliminary Facts

The main text (Definition 2.1) defines DSFs using a *layered* computation network $U_0, U_1, \ldots, U_K$ with edges directed strictly from lower to higher layers. Here we state the more general DAG form.

Definition A.1 (General DAG form of DSF)**.** A *DAG-DSF* on ground set V is defined by a directed acyclic computation graph $G = (N, E)$ with:

- *Input nodes* $V \subset N$, with $z_i(x) = x_i$ for $i \in V$.
- *Nonlinear nodes* $N \setminus V$. Each $u \in N \setminus V$ computes

$$z_u(x) \;=\; \varphi_u\left(\sum_{v:\,(v,u)\in E} w_{uv}\, z_v(x)\right), \qquad w_{uv} \geq 0,$$

  where $\varphi_u : \mathbb{R}_+ \to \mathbb{R}_+$ is concave, non-decreasing, with $\varphi_u(0) = 0$. The parents of u are any nodes v with $(v, u) \in E$, not necessarily from a single layer.
- A nonnegative linear output: $\tilde{F}(x) = \sum_{u\in\text{sinks}} a_u\, z_u(x) + \langle m_+, x\rangle$, $a_u \geq 0$, $m_+ \in \mathbb{R}^V_+$.

The *depth* of the DAG-DSF is the length of the longest directed path from an input node to a sink.

Equivalence to the layered form. Any DAG can be topologically sorted and assigned to layers $U_0, U_1, \ldots, U_K$ such that all edges go from lower to higher layers (by assigning each node to the length of its longest incoming path from an input). If the main-text layered form only allows edges between adjacent layers, an edge that skips layers can be replaced by identity pass-through nodes with activation $\varphi(t) = t$ and edge weight 1. For each output sink in a layer U_ℓ with $\ell < K$, append an identity chain to U_K and transfer its output coefficient to the terminal node. Thus every output contribution is represented in U_K, yielding a layered DSF in the sense of Definition 2.1 that computes the same function.

The layered form is therefore without loss of generality for the structural and Hessian results in this paper. Algorithmic statements still use the explicitly supplied factorization. The distinction matters primarily for *depth counting*: a DAG-DSF of depth K can be placed into $K+1$ layers by longest-path distance. Identity pass-through nodes fill skipped layers and extend shallower output sinks without changing the computed function or increasing the maximum path length K.

Effective supports. For both forms, the effective support of a nonlinear node u is defined recursively: $E(u) := \bigcup_{v:(v,u)\in E,\, w_{uv}>0} E(v)$ with $E(i) := \{i\}$ for input nodes. For a layer-1 nonlinear node in the layered form, this reduces to $E(u) = \{i \in V : w_{ui} > 0\} = \{i \in V : m_u(i) > 0\}$, matching Definition 2.2. Original nodes retain their supports under layering. Identity pass-through nodes can add duplicate supports or singleton supports, but each lies inside the support of an existing destination node or equals the support of an extended output sink. Hence the primal graph and its treewidth are unchanged. The literal indexed hyperedge family need not be identical.

B Proofs for Section 3.1: CP Hessian Characterization

B.1 Positive-interior smoothness

A nonzero normalized, nonnegative, concave non-decreasing activation is strictly positive at every positive argument. Indeed, if $\varphi(t_0) = 0$ for $t_0 > 0$, monotonicity makes it zero on $[0, t_0]$, and for $t > t_0$ concavity gives $0 = \varphi(t_0) \geq (t_0/t)\varphi(t)$, so it is identically zero.

Prune identically zero nodes and zero-weight edges. If no nonlinear nodes remain, the function is modular. At a positive input $x \in \mathbb{R}^V_{++}$, every retained input is positive. Inductively, a retained preactivation is a nonnegative sum with at least one positive term, and its nonzero activation is positive. Thus all retained preactivations lie in $(0, \infty)$, where C^2 regularity suffices for the chain rule. Deleted first-layer directions may be retained in the dictionary with coefficient zero. This includes $\sqrt{t}$ and t^α, $0 < \alpha < 1$, without differentiability at zero. Capped activations remain covered by the discrete and optimization theorems, which require no derivatives.

B.2 GRADIENT FACTORIZATION

Regrouping the DSF supergradient path expansion of Bai et al. (2018, Section 4.1, Eq. (7)) by first-layer modular directions gives this factorization in the smooth setting.

Lemma B.1 (Gradient factorization)**.** *Let F be a depth-K DSF (Definition 2.1) with normalized concave non-decreasing activations that are C^2 on $(0,\infty)$ and continuous extension $\tilde{F}:\mathbb{R}_+^V\to\mathbb{R}_+$. Let $m_1,\dots,m_{W_1}\in\mathbb{R}_+^V$ be the layer-1 modular weight vectors. Then for every $x\in\mathbb{R}_{++}^V$,*

$$\nabla\tilde{F}(x) \;=\; m_+ + \sum_{j=1}^{W_1}\alpha_j(x)\,m_j, \qquad \alpha_j(x)\ge 0,$$

where $\alpha_j(x)$ aggregates the nonnegative path products from layer-1 node j to the output in the pruned network of Section B.1, with $\alpha_j=0$ for removed layer-1 nodes.

Proof. We work with the layered computation graph of Definition 2.1. Input nodes are coordinates: $z_i(x)=x_i$ for $i\in V=U_0$. For each nonlinear node $u\in U_\ell$ ($\ell\ge 1$), write

$$g_u(x) := \sum_{v\in U_{\ell-1}} w_{uv}\,z_v(x), \qquad z_u(x) := \varphi_u(g_u(x)),$$

where $w_{uv}\ge 0$ and φ_u is concave, non-decreasing, and C^2 on $(0,\infty)$. We work on the pruned network from Section B.1. The final output is $\tilde{F}(x)=\langle m_+,x\rangle+\sum_{u\in U_K}a_u\,z_u(x)$ with $a_u\ge 0$.

For a layer-1 node j, $g_j(x)=\sum_{i\in V}w_{ji}\,x_i=\langle m_j,x\rangle$ and $\nabla z_j(x)=\varphi_j'(g_j(x))\,m_j$, a nonnegative multiple of the layer-1 modular direction $m_j\in\mathbb{R}_+^V$. For any higher-layer node $u\in U_\ell$ ($\ell\ge 2$), the chain rule gives $\nabla z_u(x)=\varphi_u'(g_u(x))\sum_{v\in U_{\ell-1}}w_{uv}\,\nabla z_v(x)$. Since $\varphi_u'\ge 0$ and $w_{uv}\ge 0$, induction over layers shows that

$$\nabla z_u(x)=\sum_{j\in U_1}\alpha_j^{(u)}(x)\,m_j, \qquad \alpha_j^{(u)}(x)\ge 0,$$

where for any node u reachable from layer-1 node j,

$$\alpha_j^{(u)}(x) \;=\; \varphi_j'(g_j(x))\sum_{P:\,j\rightsquigarrow u}\;\prod_{\text{edges }(v',v)\in P,\ v\neq j} w_{vv'}\,\varphi_v'(g_v(x)).$$

Differentiating the output $\tilde{F}=\sum_u a_u z_u+\langle m_+,x\rangle$ gives

$$\nabla\tilde{F}(x)=m_++\sum_{j\in U_1}\Big(\sum_{u\in U_K}a_u\,\alpha_j^{(u)}(x)\Big)m_j = m_+ + \sum_{j=1}^{W_1}\alpha_j(x)\,m_j,$$

where $W_1=|U_1|$ and $\alpha_j(x)=\sum_{u\in U_K}a_u\,\alpha_j^{(u)}(x)\ge 0$. This proves the claimed factorization. □

B.3 PROOF OF LEMMA 3.1

Proof. Fix $x\in\mathbb{R}_{++}^V$ and use the pruned network of Section B.1, so every retained preactivation is positive. We prove by induction on the computation graph that $-\nabla^2 z_u(x)\in\mathcal{CP}_n$ for every node u. At an input node, $z_i(x)=x_i$ and $\nabla^2 z_i(x)=0$. Suppose the claim holds for the predecessors of a nonlinear node u, and write $g_u=\sum_v w_{uv}z_v$. By the chain rule,

$$\nabla z_u=\varphi_u'(g_u)\nabla g_u, \tag{6}$$

$$\nabla^2 z_u=\varphi_u''(g_u)\nabla g_u\nabla g_u^\top+\varphi_u'(g_u)\nabla^2 g_u. \tag{7}$$

Every node is coordinatewise non-decreasing, so $\nabla g_u\ge 0$. Consequently $\nabla g_u\nabla g_u^\top\in\mathcal{CP}_n$. Moreover, $-\varphi_u''(g_u)\ge 0$ and $\varphi_u'(g_u)w_{uv}\ge 0$. The induction hypothesis and closure of $\mathcal{CP}_n$ under nonnegative sums therefore give

$$-\nabla^2 z_u=-\varphi_u''(g_u)\nabla g_u\nabla g_u^\top+\sum_v\varphi_u'(g_u)w_{uv}(-\nabla^2 z_v)\in\mathcal{CP}_n.$$

Finally, the modular term has zero Hessian, and hence

$$-\nabla^2\tilde{F}(x)=\sum_{u\in U_K}a_u(-\nabla^2 z_u(x))\in\mathcal{CP}_n.$$

□

B.4 RIDGE-DICTIONARY SPAN OF THE HESSIAN

Proposition B.2 (Ridge-dictionary span). *Under the hypotheses of Lemma B.1, for every $x \in \mathbb{R}^V_{++}$,*

$$\operatorname{image}(\nabla^2 \tilde{F}(x)) \subseteq \operatorname{span}(m_1, \dots, m_{W_1}) \quad \textit{and} \quad \operatorname{rank}(\nabla^2 \tilde{F}(x)) \leq W_1.$$

Proof. From the proof of Lemma 3.1, each $-\nabla^2 z_u(x)$ is a sum of terms of the form $\lambda\, vv^\top$ where $v \in \mathbb{R}^n_+$ lies in the conic span of $\{m_1, \dots, m_{W_1}\}$. Therefore $v \in \operatorname{span}(m_1, \dots, m_{W_1})$, and the image of $vv^\top$ is $\operatorname{span}(v) \subseteq \operatorname{span}(m_1, \dots, m_{W_1})$. Since $-\nabla^2 \tilde{F}(x)$ is a nonnegative combination of such rank-one matrices, its image lies in $\operatorname{span}(m_1, \dots, m_{W_1})$, giving $\operatorname{rank}(\nabla^2 \tilde{F}(x)) \leq W_1$. □

C TRACE-CYCLE LEMMA

Lemma C.1 (Trace lower bound on C_5-supported CP matrices). *If $M \in \mathcal{CP}_5$ has support contained in the 5-cycle C_5 (i.e. $M_{ij} = 0$ for non-adjacent $i \neq j$), then*

$$\sum_i M_{ii} \geq 2 \sum_{(i,j) \in E(C_5)} M_{ij}.$$

Proof. Write $M = \sum_{\ell=1}^r v_\ell v_\ell^\top$ with $v_\ell \in \mathbb{R}^5_+$. For a nonedge $\{i, j\}$ of C_5,

$$0 = M_{ij} = \sum_{\ell=1}^r (v_\ell)_i (v_\ell)_j.$$

All summands are nonnegative, so $(v_\ell)_i (v_\ell)_j = 0$ for every ℓ. Thus the support of each v_ℓ is a clique of C_5 and has at most two vertices. If its two possibly nonzero entries are $a, b \geq 0$, then $\|v_\ell\|^2 = a^2 + b^2 \geq 2ab$. This also covers a singleton or empty support by allowing one or both entries to be zero. Summing gives

$$\sum_{i=1}^5 M_{ii} = \sum_{\ell=1}^r \|v_\ell\|^2 \geq 2 \sum_{\ell=1}^r \sum_{\{i,j\} \in E(C_5)} (v_\ell)_i (v_\ell)_j = 2 \sum_{\{i,j\} \in E(C_5)} M_{ij}.$$

□

D CONTINUOUS-EXTENSION EXPRESSIVITY GAP

D.1 A FIVE-CYCLE MATRIX WITNESS

For the matrix H in (3), the circulant eigenvalues are $5 + 6\cos(2\pi j/5)$, $j = 0, \dots, 4$. Their minimum is $(7 - 3\sqrt{5})/2 > 0$. Since all entries are nonnegative, $H \in \mathcal{DNN}_5$. Every C_5-supported CP matrix satisfies $\sum_i M_{ii} \geq 2 \sum_{ij \in E(C_5)} M_{ij}$ (Section C). The diagonal and edge sums for H are 25 and 15, respectively, so $H \notin \mathcal{CP}_5$.

D.2 SATURATING-MAP LEMMA

Lemma D.1 (Saturating map). *For every $x_0 > 0$, there exists a C^∞ function $\varphi : \mathbb{R}_+ \to \mathbb{R}_+$ satisfying:*

(i) $\varphi(0) = 0$.

(ii) φ *is non-decreasing:* $\varphi'(t) \geq 0$ *for all* $t \geq 0$.

(iii) φ *is concave:* $\varphi''(t) \leq 0$ *for all* $t \geq 0$.

(iv) φ *is bounded:* $L := \lim_{t \to \infty} \varphi(t) < \infty$.

(v) $\varphi'(x_0) = 1$ *and* $\varphi''(x_0) = 0$.

Proof. Fix $b > x_0$ and $\delta > 0$. Choose a C^∞ non-increasing cutoff $p : \mathbb{R}_+ \to [0,1]$ such that $p(t) = 1$ for $t \le b$, $p(t) = 0$ for $t \ge b + \delta$, and $p'(t) \le 0$. (Such a cutoff exists by standard mollifier constructions.) Define

$$\varphi(t) \;:=\; \int_0^t p(s)\, ds.$$

Then $\varphi(0) = 0$. $\varphi'(t) = p(t) \in [0,1]$, so φ is non-decreasing. $\varphi''(t) = p'(t) \le 0$, so φ is concave. Finally, $L := \lim_{t\to\infty} \varphi(t) = \int_0^{b+\delta} p(s)\, ds < \infty$, so φ is bounded. Since $x_0 < b$ and $p \equiv 1$ on $[0, b]$, we have $\varphi'(x_0) = p(x_0) = 1$ and $\varphi''(x_0) = p'(x_0) = 0$. □

D.3 Proof of Theorem 3.2

Proof. Fix $n \ge 5$, $B \in \mathcal{DNN}_n \setminus \mathcal{CP}_n$, and $x_0 \in (0,\infty)^n$. For each i, choose φ_i from Lemma D.1 with $\varphi_i'(x_{0,i}) = 1$ and $\varphi_i''(x_{0,i}) = 0$, and let $L_i = \sup_{t\ge 0} \varphi_i(t) < \infty$. Set

$$\phi(x) = (\varphi_1(x_1), \dots, \varphi_n(x_n)), \qquad c_i = \sum_{j=1}^n b_{ij} L_j,$$

and define

$$F(x) = \sum_{i=1}^n c_i \varphi_i(x_i) - \frac{1}{2}\phi(x)^\top B \phi(x).$$

The function F is C^∞ and satisfies $F(0) = 0$. Since B has nonnegative entries and $\phi_j(x) \le L_j$, differentiation gives

$$\frac{\partial F}{\partial x_i} = \varphi_i'(x_i)\big(c_i - (B\phi(x))_i\big) \ge 0.$$

Thus F is coordinatewise non-decreasing and takes values in $\mathbb{R}_+$.

Let $D(x) = \operatorname{diag}(\varphi_1'(x_1), \dots, \varphi_n'(x_n))$. A second differentiation yields

$$\nabla^2 F(x) = \operatorname{diag}\big(\varphi_i''(x_i)\big(c_i - (B\phi(x))_i\big)\big) - D(x)BD(x).$$

The diagonal term is negative semidefinite because $\varphi_i'' \le 0$ and $c_i - (B\phi(x))_i \ge 0$. Since $B \succeq 0$, the matrix $D(x)BD(x)$ is positive semidefinite. Hence $\nabla^2 F(x) \preceq 0$, proving concavity. For $i \ne j$,

$$\frac{\partial^2 F}{\partial x_i \partial x_j} = -b_{ij}\varphi_i'(x_i)\varphi_j'(x_j) \le 0, \tag{8}$$

which proves continuous lattice-submodularity.

At x_0, the choice of the maps gives $D(x_0) = I$ and $\varphi_i''(x_{0,i}) = 0$ for every i. Therefore $-\nabla^2 F(x_0) = B \notin \mathcal{CP}_n$. By Lemma 3.1, F cannot be a DSF continuous extension with activations that are C^2 on $(0,\infty)$. □

Remark D.2 (Dimension threshold). For $n \le 4$, $\mathcal{CP}_n = \mathcal{DNN}_n$ (Berman & Shaked-Monderer, 2003). Thus the separation by a matrix in $\mathcal{DNN}_n \setminus \mathcal{CP}_n$ requires $n \ge 5$. This condition concerns the Hessian obstruction used here.

Example D.3 (Five-cycle witness)**.** For the matrix H in Section D.1, every row sum equals 11. Taking $L = \max_i L_i$ and $c_i = 11L$ in the preceding construction gives $-\nabla^2 F(x_0) = H$. The matrix H is doubly nonnegative, but

$$\sum_i H_{ii} = 25 < 30 = 2 \sum_{\{i,j\}\in E(C_5)} H_{ij},$$

so Lemma C.1 implies $H \notin \mathcal{CP}_5$.

The construction excludes a specified continuous extension. A set function with the same cube trace could have another DSF extension, so Theorem 3.5 establishes the discrete separation independently.

D.4 SUPPORT-SENSITIVE CURVATURE SEPARATION

Proof of Corollary 3.3. The graph of a symmetric matrix B has edge ij exactly when $i \neq j$ and $B_{ij} \neq 0$. Kogan–Berman's characterization states that every doubly nonnegative matrix with graph G is completely positive if and only if G has no odd cycle of length at least five (Kogan & Berman, 1993). If G has such a cycle, take $B \in \mathcal{DNN}_n \setminus \mathcal{CP}_n$ with that graph and apply Theorem 3.2. By (8), $B_{ij} = 0$ implies $\partial_{ij} F(x) = 0$ for all $x \in \mathbb{R}^V_+$. Hence the negative Hessian is G-supported throughout the domain and equals B at x_0. By Lemma 3.1, this excludes a DSF extension with $C^2((0,\infty))$ activations.

Conversely, let G have no such odd cycle. A negative Hessian of a smooth concave lattice-submodular function is positive semidefinite and entrywise nonnegative. If it is G-supported, its graph is a subgraph of G and also has no such cycle. The characterization implies complete positivity. □

E ODD-CYCLE COMPUTATIONAL EVIDENCE

Proposition E.1 (Odd-cycle separation for a finite atom dictionary)**.** *The paving rank-3 matroid on $V = \{0,1,2,3,4,5\}$ defined below is not expressible as a nonnegative combination of atoms from the 192-element dictionary $\{\min(|\cdot \cap S|, k) : S \subseteq V,\ 1 \le k \le |S|\}$. This rules out depth-1 sums of concave cardinality-on-support factors $\varphi(|A \cap S|)$ (unit weights on each support), including nonnegative piecewise-linear activations with integer breakpoints, but does not by itself rule out all depth-1 DSFs with arbitrary concave activations or all multi-depth DSFs. The all-depth discrete separation is Theorem 3.5.*

Proof. Let $C_1 = \{0,1,3\}$, $C_2 = \{1,2,4\}$, and $C_3 = \{0,2,5\}$. The bases are all three-element subsets of V other than C_1, C_2, C_3. These sets satisfy basis exchange. Indeed, for bases B_1, B_2 and $e \in B_1 \setminus B_2$, if $|B_2 \setminus B_1| = 1$, exchanging e gives B_2. Otherwise, the candidate exchanges share the pair $B_1 \setminus \{e\}$. At most one candidate can equal a C_i, since distinct C_i intersect in only one element. Thus at least one candidate is a basis. Every pair extends to a basis, and the resulting rank function is

$$r(A) = \begin{cases} |A| & |A| \le 2, \\ 2 & A \in \{C_1, C_2, C_3\}, \\ 3 & \text{otherwise.} \end{cases}$$

In particular, r is normalized, monotone, and submodular.

Index subsets by $\iota(A) = \sum_{i \in A} 2^i$ and define $\lambda_A = 1$ when $\iota(A) \in P$, $\lambda_A = -1$ when $\iota(A) \in N$, and zero otherwise, where

$$\begin{aligned} P = \{&5, 9, 14, 15, 18, 19, 25, 26, 27, 28, 31, 33, \\ &38, 39, 41, 42, 43, 50, 51, 52, 53, 56, 58\}, \\ N = \{&0, 1, 4, 10, 11, 13, 20, 22, 23, 24, 30, 34, 35, \\ &37, 40, 45, 46, 47, 48, 49, 54, 55, 59, 61, 62, 63\}. \end{aligned}$$

Direct integer evaluation gives

$$\sum_{A \subseteq V} \lambda_A r(A) = 3, \qquad \sum_{A \subseteq V} \lambda_A \min(|A \cap S|, k) \le 0 \quad (S \subseteq V,\ 1 \le k \le |S|).$$

These are 192 finite inequalities. If r were a nonnegative combination of the dictionary atoms, multiplying these inequalities by its coefficients and summing would give $\sum_A \lambda_A r(A) \le 0$, a contradiction.

To include all normalized concave cardinality factors, let $d_j = \varphi(j) - \varphi(j-1)$ for $1 \le j \le m = |S|$, and set $d_{m+1} = 0$. Concavity and monotonicity give $d_j - d_{j+1} \ge 0$, and

$$\varphi(t) = \sum_{j=1}^{m} (d_j - d_{j+1}) \min(t, j) \qquad (t = 0, \dots, m).$$

Thus every such factor belongs to the same dictionary cone. □

F K_4 CLIQUE-OVERCOUNT PRESERVATION AND HINGE LEMMA

F.1 PRESERVATION PROOF

We prove that every DSF cube trace on $V = E(K_4)$ satisfies Property P (Definition 3.4). The proof proceeds by structural induction on the DSF computation.

Recall: $\rho_f(S) = \sum_{e \in S} f(\{e\}) - f(S)$, and Property P states that if $\rho_f(\{e, e'\}) = 0$ for all $e \neq e'$ (pair-additivity), then $\rho_f(V) \geq \sum_{T \in \mathcal{T}} \rho_f(T)$, where $\mathcal{T} = \{T_1, T_2, T_3, T_4\}$ are the four triangles of K_4.

Proof of Theorem 3.5, preservation. We show that Property P is preserved under each DSF operation.

If f is modular, then $f(S) = \sum_{e \in S} f(\{e\})$ for all S, so $\rho_f(S) = 0$ for every S. In particular, $\rho_f(V) = 0 = \sum_T \rho_f(T)$. Property P holds (the conclusion is $0 \geq 0$).

Let $f = \sum_i f_i$, absorbing nonnegative coefficients into normalized monotone submodular summands f_i, each satisfying Property P. Since $\rho_f = \sum_i \rho_{f_i}$ (deficit is linear in f), pair-additivity of f means $\sum_i \rho_{f_i}(\{e, e'\}) = 0$ for all pairs. Each f_i is submodular, so $\rho_{f_i}(\{e, e'\}) \geq 0$ (subadditivity on pairs). Since the sum of nonnegative terms is zero, each term vanishes: $\rho_{f_i}(\{e, e'\}) = 0$ for all i and all pairs $e \neq e'$.

Therefore each f_i is pair-additive, and Property P applies to each:

$$\rho_f(V) = \sum_i \rho_{f_i}(V) \geq \sum_i \sum_{T \in \mathcal{T}} \rho_{f_i}(T) = \sum_{T \in \mathcal{T}} \rho_f(T).$$

Let h be normalized, monotone, submodular, and satisfy Property P, and let φ be concave and non-decreasing with $\varphi(0) = 0$. Define $f = \varphi \circ h$. Suppose f is pair-additive. We show $I(f) := \rho_f(V) - \sum_T \rho_f(T) \geq 0$.

If $\varphi \equiv 0$, the conclusion is immediate. Otherwise, fix distinct e, e' and write $a = h(\{e\})$, $b = h(\{e'\})$, and $c = h(\{e, e'\})$. Monotonicity and submodularity give $\max(a, b) \leq c \leq a + b$. If $a = 0$ or $b = 0$, these bounds imply $c = a + b$. For $a, b > 0$, concavity and $\varphi(0) = 0$ give

$$\varphi(c) \leq \varphi(a + b) \leq \varphi(a) + \varphi(b) = \varphi(c).$$

Equality in the concavity inequalities at the interior points a, b forces φ to be linear on $[0, a + b]$. Its slope is positive because a nonzero, nonnegative, non-decreasing concave function vanishing at zero is positive on $(0, \infty)$. Hence $\varphi(c) = \varphi(a + b)$ implies $c = a + b$. Therefore h is pair-additive and $I(h) \geq 0$ by Property P. The activation reduction in Section F.3 then gives $I(f) \geq 0$.

□

F.2 HINGE LEMMA PROOF

Lemma F.1 (Hinge lemma)**.** *Let $h : 2^V \to \mathbb{R}_+$ with $V = E(K_4)$ be normalized, monotone submodular, pair-additive, and satisfy $I(h) := \rho_h(V) - \sum_{T \in \mathcal{T}} \rho_h(T) \geq 0$. Then for every $s \geq \tau := \max_{e \neq e'}(h(\{e\}) + h(\{e'\}))$,*

$$I(h) + J_s(h) \geq 0, \qquad J_s(h) := (h(V) - s)_+ - \sum_{T \in \mathcal{T}} (h(T) - s)_+.$$

Proof. Write $a_e := h(\{e\})$, $A(S) := \sum_{e \in S} a_e$, $r_S := \rho_h(S) = A(S) - h(S)$ (so $h(S) = A(S) - r_S$). Pair-additivity gives $h(\{e, e'\}) = a_e + a_{e'}$ and $r_{\{e,e'\}} = 0$. The key bounds we will use:

- *Pair sum bound.* $a_e + a_{e'} \leq \tau \leq s$ for every $e \neq e'$. Consequently, in any set containing at least two edges, at least one edge has weight at most $s/2$.
- *Triangle redundancy bound.* For any triangle $T = \{e_1, e_2, e_3\}$ and any $e_i \in T$, monotonicity gives $h(T) \geq h(T \setminus \{e_i\}) = A(T) - a_{e_i}$ (the second equality is pair-additivity), so $r_T = A(T) - h(T) \leq a_{e_i}$. Taking the minimum: $r_T \leq \min_{e \in T} a_e$.

Let $\mathcal{A} := \{T \in \mathcal{T} : h(T) > s\}$ and $m := |\mathcal{A}|$.

If $m = 0$, then $J_s(h) = (h(V) - s)_+ \geq 0$ and $I(h) \geq 0$, so $I + J_s \geq 0$.

If $m \geq 1$, then some $T \in \mathcal{A}$ has $h(T) > s$, and monotonicity gives $h(V) \geq h(T) > s$, so $(h(V) - s)_+ = h(V) - s$. Substituting $h(V) = A(V) - r_V$, $h(T) = A(T) - r_T$, and using $\sum_T A(T) = 2A(V)$ (each edge in two triangles), one computes

$$I(h) + J_s(h) \;=\; A(V) + (m-1)s - \textstyle\sum_{T\in\mathcal{A}} A(T) \;-\; \sum_{T\notin\mathcal{A}} r_T.$$

It remains to show this is ≥ 0 in each case $m \in \{1, 2, 3, 4\}$.

If $m = 4$, then $\sum_{T\in\mathcal{A}} A(T) = \sum_T A(T) = 2A(V)$, no inactive triangles, so the expression is $A(V) + 3s - 2A(V) = 3s - A(V)$. The six edges of K_4 can be grouped into three disjoint two-edge pairs, each with weight sum $\leq \tau \leq s$, so $A(V) \leq 3s$ and $3s - A(V) \geq 0$.

If $m = 3$, then let $T_0 \notin \mathcal{A}$. Then $\sum_{T\in\mathcal{A}} A(T) = 2A(V) - A(T_0)$, and the expression is

$$A(V) + 2s - 2A(V) + A(T_0) - r_{T_0} \;=\; 2s - A(C) - r_{T_0},$$

where $C := V \setminus T_0$ has 3 edges. The three pairs in C each have weight sum $\leq s$, summing to $2A(C) \leq 3s$, so $A(C) \leq 3s/2$. Triangle redundancy on T_0 gives $r_{T_0} \leq \min_{e\in T_0} a_e$. For any two distinct edges $e, e' \in T_0$, the pair $\{e, e'\}$ satisfies $a_e + a_{e'} \leq \tau \leq s$, so $2\min_{e\in T_0} a_e \leq a_e + a_{e'} \leq s$, yielding $\min_{e\in T_0} a_e \leq s/2$. Total $\leq 3s/2 + s/2 = 2s$.

If $m = 2$, then let the active triangles be T_1, T_2 with shared edge p, and the inactive triangles U_1, U_2. In K_4, any two triangles share exactly one edge. The unique edge of V contained in neither T_1 nor T_2 is q, and q belongs to both U_1, U_2. One computes $\sum_{T\in\mathcal{A}} A(T) = A(T_1) + A(T_2) = A(V) + a_p - a_q$, so the expression is

$$A(V) + s - (A(V) + a_p - a_q) - r_{U_1} - r_{U_2} \;=\; s - a_p + a_q - r_{U_1} - r_{U_2}.$$

The triangle redundancy bound gives $r_{U_j} \leq a_q$ for each j (since $q \in U_j$), so $r_{U_1} + r_{U_2} \leq 2a_q$. Substituting,

$$s - a_p + a_q - 2a_q \;=\; s - a_p - a_q \;\geq\; 0$$

because $a_p + a_q \leq \tau \leq s$ is one of the pair-sum bounds.

If $m = 1$, then let $\mathcal{A} = \{T_0\}$ and $C := V \setminus T_0$. The expression is

$$A(V) + 0 \cdot s - A(T_0) - \textstyle\sum_{T\neq T_0} r_T \;=\; A(C) - \sum_{T\neq T_0} r_T.$$

The three inactive triangles each contain exactly one edge of T_0 and a 2-subset of C. Label $C = \{c_1, c_2, c_3\}$. Then for distinct $i, j \in \{1, 2, 3\}$, $\{c_i, c_j\} \subseteq U_{ij}$ for a unique inactive triangle U_{ij}. By the triangle redundancy bound, $r_{U_{ij}} \leq \min(a_{c_i}, a_{c_j})$. Hence

$$\textstyle\sum_{T\neq T_0} r_T \;\leq\; \min(a_{c_1}, a_{c_2}) + \min(a_{c_1}, a_{c_3}) + \min(a_{c_2}, a_{c_3}).$$

For nonnegative x_1, x_2, x_3 (assume WLOG $x_1 \leq x_2 \leq x_3$), $\min(x_1, x_2) + \min(x_1, x_3) + \min(x_2, x_3) = x_1 + x_1 + x_2 = 2x_1 + x_2 \leq x_1 + x_2 + x_3$, so the right-hand side is $\leq A(C)$. Hence $A(C) - \sum_{T\neq T_0} r_T \geq 0$.

This completes the case analysis and the proof of Lemma F.1. □

F.3 General activation reduction

We show how the hinge lemma gives $I(\varphi \circ h) \geq 0$ for any concave non-decreasing activation φ with $\varphi(0) = 0$, given that h is normalized, monotone, submodular, pair-additive, satisfies $I(h) \geq 0$, and $\varphi \circ h$ is pair-additive.

Proof. Write $g = \varphi \circ h$. Let $a_e := h(\{e\})$ and $\tau := \max_{e\neq e'}(a_e + a_{e'})$.

If at most one a_e is positive, then all nonzero values of h are determined by that singleton and $I(\varphi \circ h) = 0$ trivially. Indeed, subadditivity gives $h(S) \leq \sum_{e\in S} a_e$, while monotonicity gives equality whenever the unique positive singleton is present. Hence every triangle and global deficit is zero. Otherwise, take a pair attaining τ with both endpoints positive. By hypothesis, h is pair-additive:

$h(\{e, e'\}) = a_e + a_{e'}$ for all pairs. The equality throughout the subadditivity chain (specifically, $\varphi(a_e) + \varphi(a_{e'}) = \varphi(a_e + a_{e'})$ for a pair with $a_e, a_{e'} > 0$) forces φ to be linear on $[0, \tau]$: $\varphi(t) = \alpha t$ for $t \in [0, \tau]$ and some $\alpha \geq 0$.

Define $\psi(t) := \alpha t - \varphi(t)$. Then:

- ψ is convex (since αt is linear and $-\varphi$ is convex).
- ψ is non-decreasing: concavity of φ and linearity on $[0, \tau]$ with slope α imply $D^+\varphi(t) \leq \alpha$ for $t \geq \tau$ (where D^+ denotes the right derivative), hence $D^+\psi(t) = \alpha - D^+\varphi(t) \geq 0$.
- $\psi \equiv 0$ on $[0, \tau]$ (since $\varphi(t) = \alpha t$ there).
- $D^+\psi(t) \leq \alpha$ everywhere (since $D^+\varphi(t) \geq 0$).

Since all singleton values satisfy $a_e \leq \tau$, we have $\varphi(a_e) = \alpha a_e$, giving the redundancy decomposition

$$\rho_g(S) \;=\; \alpha\, \rho_h(S) \;+\; \psi(h(S)). \tag{9}$$

Indeed, $\rho_g(S) = \sum_e \varphi(a_e) - \varphi(h(S)) = \alpha \sum_e a_e - \varphi(h(S)) = \alpha[A(S) - h(S)] + [\alpha h(S) - \varphi(h(S))] = \alpha\, \rho_h(S) + \psi(h(S))$.

Define $I(f) := \rho_f(V) - \sum_T \rho_f(T)$ and $J_s(h) := (h(V) - s)_+ - \sum_T (h(T) - s)_+$. From (9),

$$I(g) \;=\; \alpha\, I(h) \;+\; \big[\psi(h(V)) - \textstyle\sum_T \psi(h(T))\big].$$

By hypothesis $I(h) \geq 0$.

Since ψ is convex, non-decreasing, vanishes on $[0, \tau]$, and has right derivative $D^+\psi \leq \alpha$, it admits the integral representation

$$\psi(t) \;=\; \int_\tau^\infty (t - s)_+ \, d\mu(s)$$

for a nonnegative measure μ on $[\tau, \infty)$ satisfying $\int d\mu = \sup D^+\psi \leq \alpha$. This is a standard consequence of the integral representation of convex functions via one-sided hinges (Rockafellar, 1970): $(t - s)_+$ is convex in t, and the integral starts from τ because $\psi \equiv 0$ on $[0, \tau]$. This representation requires no smoothness of φ.

Substituting the integral representation:

$$\psi(h(V)) - \textstyle\sum_T \psi(h(T)) \;=\; \int_\tau^\infty \big[(h(V) - s)_+ - \sum_T (h(T) - s)_+\big]\, d\mu(s) \;=\; \int_\tau^\infty J_s(h)\, d\mu(s).$$

By the hinge lemma (Lemma F.1), $I(h) + J_s(h) \geq 0$ for every $s \geq \tau$, so $J_s(h) \geq -I(h)$. Integrating:

$$\int_\tau^\infty J_s(h)\, d\mu(s) \;\geq\; -I(h) \int_\tau^\infty d\mu(s) \;\geq\; -\alpha\, I(h).$$

$$I(g) \;\geq\; \alpha\, I(h) \;-\; \alpha\, I(h) \;=\; 0.$$

Concave composition preserves the overcount inequality. □

F.4 Proof of Corollary 3.6

We first establish the closure property needed to pass to a matroid minor.

Proposition F.2 (Closure under restriction and normalized contraction)**.** *Let $C, W \subseteq V$ be disjoint. If F is a DSF on V, then*

$$F_C(A) = F(A \cup C) - F(C), \qquad A \subseteq W,$$

is a DSF on W.

Proof. Let z_u denote the node functions in a DSF representation of F. For each nonlinear node u, define

$$b_u = \sum_v w_{uv} z_v(\mathbf{1}_C), \qquad \psi_u(t) = \varphi_u(b_u + t) - \varphi_u(b_u).$$

The function ψ_u is normalized, nonnegative, concave, and non-decreasing on $\mathbb{R}_+$. Use the original weights with these activations, and set the new input functions to $\widehat{z}_i(x) = x_i$ for $i \in W$ and $\widehat{z}_i(x) = 0$ for $i \notin W$. We claim that for every node u and every $A \subseteq W$,

$$\widehat{z}_u(\mathbf{1}_A) = z_u(\mathbf{1}_{A\cup C}) - z_u(\mathbf{1}_C).$$

This holds at the input nodes. If it holds at the predecessors of u, then

$$\begin{aligned}\widehat{z}_u(\mathbf{1}_A) &= \psi_u\left(\sum_v w_{uv}\widehat{z}_v(\mathbf{1}_A)\right)\\ &= \varphi_u\left(\sum_v w_{uv} z_v(\mathbf{1}_{A\cup C})\right) - \varphi_u\left(\sum_v w_{uv} z_v(\mathbf{1}_C)\right)\\ &= z_u(\mathbf{1}_{A\cup C}) - z_u(\mathbf{1}_C).\end{aligned}$$

Induction proves the claim. Applying the original nonnegative output coefficients and restricting the modular term to W gives

$$\sum_{u\in U_K} a_u \widehat{z}_u(\mathbf{1}_A) + \sum_{i\in A} m_+(i) = F(A\cup C) - F(C).$$

All weights are nonnegative and all activations satisfy the DSF definition. Taking $C = \emptyset$ gives closure under restriction. □

Proof of Corollary 3.6. Suppose that M has an $M(K_4)$ minor and that r_M is a DSF cube trace. Matroid deletion restricts the rank function to a smaller ground set, and contraction by C replaces it by

$$r_{M/C}(A) = r_M(A\cup C) - r_M(C).$$

By Proposition F.2, applying the deletions and contractions that produce the minor would give a DSF representation of $r_{M(K_4)}$. This contradicts Theorem 3.5.

More generally, the same conclusion holds for $r_M + m$, where m is nonnegative modular. Under restriction and normalized contraction, m remains nonnegative modular. The resulting representation on $E(K_4)$ would therefore equal $r_{M(K_4)} + m'$ for a nonnegative modular function m'. Since $\rho_{r+m'} = \rho_r$, its pair deficits vanish, its global deficit is 3, and the sum of its triangle deficits is 4. It violates Property P, again contradicting Theorem 3.5. □

G Sub-Category Containment Relations

Definition G.1 (Representation classes)**.** For a supplied depth-1 representation, call its factor-support family partition if its distinct scopes form at least two disjoint blocks, laminar if any two scopes are nested or disjoint, and chordal if its support primal graph is chordal. Write $\mathfrak{R}_{\mathrm{Part}}, \mathfrak{R}_{\mathrm{Lam}}, \mathfrak{R}_{\mathrm{Chord}}$ for these representation classes and $\mathfrak{R}_{\mathrm{TW}(w)}$ for supplied representations of width at most w.

Theorem G.2 (Representation-level taxonomy)**.** *For depth-1 representations,* $\mathfrak{R}_{\mathrm{Part}} \subsetneq \mathfrak{R}_{\mathrm{Lam}} \subsetneq \mathfrak{R}_{\mathrm{Chord}} \subsetneq \mathfrak{R}_{\mathrm{SCMM}}$. *Chordal and width-$w$ representation classes are incomparable for $w \geq 2$. For $w \leq 1$, width-w primal graphs are chordal.*

Proof. Partition scopes are laminar. Maximal laminar scopes are disjoint cliques, so their primal graph is chordal. Nested scopes $\{1,2\} \subset \{1,2,3\}$ give the first strictness. Crossing scopes $\{1,2,3\}$ and $\{3,4,5\}$ give a chordal graph (two triangles sharing a vertex), but are not laminar. Unit-cap factors on the four edges of C_4 give a nonchordal SCMM representation of width 2. Conversely, one nonlinear ridge on $w+2$ elements gives a chordal clique of width $w+1$. These are statements about supplied representations, not exclusions of all alternative representations of the same function. □

An all-depth function-level separation. Here a partition DSF means a representation whose distinct nonempty effective supports partition V into at least two blocks. A laminar DSF has nested-or-disjoint effective supports. Partition DSFs form a proper subclass of laminar DSFs even at the cube-trace level.

Proof. Take $V = V_1 \sqcup V_2 \sqcup V_3$ with $|V_i| = 3$ (so $|V| = 9$), and define

$$f(A) \;:=\; \min\Big(\textstyle\sum_{i=1}^{3} \min(|A \cap V_i|, 2),\; 4\Big).$$

This is the rank function of the laminar matroid with per-class cap $k_i = 2$ and root cap $k = 4 < 6 = \sum k_i$, hence a depth-2 DSF whose effective clique hypergraph $\{V_1, V_2, V_3, V\}$ is laminar. We show f is not the cube trace of any DSF with partition effective clique structure.

Suppose $f = g$ for some DSF whose effective clique hypergraph is a partition $D_1, \ldots, D_p$ with $p \geq 2$. If a nonlinear node u had children from distinct blocks D_i, D_j, then $E(u)$ would intersect both blocks, contradicting its containment in a single maximal effective support. Hence the computation decomposes additively across blocks: $g(A) = \sum_{r=1}^{p} g_r(A \cap D_r)$, giving $g(V) = \sum_r g(D_r)$.

We show $f(D) + f(V \setminus D) > f(V)$ for every nonempty proper $D \subset V$, which contradicts $g(V) = \sum_r g(D_r)$ (since for $p \geq 2$, summing over blocks gives $\sum_r f(D_r) \geq f(D_1) + f(V \setminus D_1) > f(V)$). Write $a_i = |D \cap V_i|$. Let $h(A) = \sum_{i=1}^{3} \min(|A \cap V_i|, 2)$. For each group V_i, $\min(a_i, 2) + \min(3 - a_i, 2) \geq 2$ (the sum is 2 for $a_i \in \{0, 3\}$ and 3 otherwise). Summing over i: $h(D) + h(V \setminus D) \geq 6$. Since D and $V \setminus D$ are nonempty, $h(D) \geq 1$ and $h(V \setminus D) \geq 1$, so $f(D) + f(V \setminus D) = \min(h(D), 4) + \min(h(V \setminus D), 4) \geq 5 > 4 = f(V)$.

Combined with $f \in$ Laminar-DSF by construction, this establishes Partition-DSF $\subsetneq$ Laminar-DSF.

□

Interpretation. Representation taxonomy alone does not prove strict containments of the corresponding function classes, because the same function can admit other factorizations. Nor does coverage hardness exclude arbitrary exponential-size bounded-width representations. Example 4.5 supplies a separate intrinsic lower bound for additive factorizations of its particular function. For deep computation trees, disjoint child supports imply laminar effective supports (Section N). Laminar supports alone are not our algorithmic hypothesis.

H Junction-Tree DP: Correctness and Complexity

The complete pseudocode for the junction-tree DP described in Section 4.1 is given below.

H.1 Correctness proof

We prove that Algorithm 2 computes $\max_{S \subseteq V,\, |S| \leq k} F(S)$ exactly.

Proof of Theorem 4.1, correctness. First, $z_u(x)$ depends only on $x|_{E(u)}$. For $u \in U_1$, this follows from $z_u(x) = \varphi_u(\langle m_u, x \rangle)$ and $E(u) = \mathrm{supp}(m_u)$. Inductively, every predecessor v with $w_{uv} > 0$ satisfies $E(v) \subseteq E(u)$, so the same claim holds for $z_u = \varphi_u(\sum_v w_{uv} z_v)$. Hence $z_u(\mathbf{1}_A) = z_u(\mathbf{1}_{A \cap E(u)})$, proving (4). The supplied decomposition covers every effective support, including those of the top-layer nodes. Each f_u can therefore be assigned to a bag containing $E(u)$, with $|E(u)| \leq w + 1$. Shared subcomputations do not affect this additive decomposition. We now induct on the rooted tree decomposition.

Fix a rooted tree decomposition $\mathcal{T} = (T, \{B_t\}_{t \in T})$ of the effective clique hypergraph $\mathcal{H}_F^{\mathrm{eff}}$. For each non-root node t, write $S_t = B_t \cap B_{\mathrm{par}(t)}$ for the separator and $R_t = B_t \setminus S_t$ for the private variables. Let T_t denote the subtree of T rooted at t, and let $V_t^* = \bigcup_{s \in T_t} R_s$ be the set of all private variables in the subtree. Set $S_r = \emptyset$ and $R_r = B_r$ at the root. The running-intersection property gives $V_t^* \cap S_t = \emptyset$ and $\bigcup_{s \in T_t} B_s = V_t^* \sqcup S_t$. Every variable in V_t^* appears only in bags of T_t.

For each bag t, assignment $a_t \subseteq B_t$, and budget $b \in \{0, \ldots, k\}$, the table entry $V_t(a_t, b)$ computed by Algorithm 2 equals

$$V_t(a_t, b) \;=\; \max\Big\{ \sum_{h \in \mathcal{F}_s,\, s \in T_t} h(A \cap E(h)) \;:\; A \subseteq V_t^* \cup S_t,\; A \cap B_t = a_t,\; |A \cap V_t^*| = b \Big\},$$

Algorithm 2 Junction-tree DP for cardinality-constrained DSF maximization

Input: DSF $F(S) = \sum_{u \in U_K} f_u(S \cap E(u)) + \langle m_+, \mathbf{1}_S \rangle$ with top-layer factor tables as in (4). Budget k. Tree decomposition $\mathcal{T} = (T, \{B_t\}_{t \in T})$ of $\mathcal{H}_F^{\text{eff}}$ (width w).
Output: Exact optimum $\max_{S \subseteq V, |S| \le k} F(S)$.

Preprocess (tail): for each i with $m_+(i) > 0$, add the unary factor $g_i(S) = m_+(i)\,\mathbf{1}[i \in S]$ (support $\{i\}$) to the factor family.
Root T at an arbitrary node r.
for each non-root t **do**
    $S_t \leftarrow B_t \cap B_{\text{par}(t)}, \quad R_t \leftarrow B_t \setminus S_t.$ ▷ separator, private vars
end for
$S_r \leftarrow \emptyset, \; R_r \leftarrow B_r.$
Assign each factor (top-layer f_u or unary g_i) to one bag t containing its support. Call the assigned set $\mathcal{F}_t$.
for bags $t \in T$ in **postorder do**
    $\Phi_t(a_t) \leftarrow \sum_{h \in \mathcal{F}_t} h(a_t \cap E(h))$ for each $a_t \subseteq B_t$. ▷ local factors, incl. unary tail
    if t is a leaf **then**
        $V_t(a_t, b) \leftarrow \Phi_t(a_t)$ if $b = |a_t \cap R_t|$, else $-\infty$.
    else
        for each child c of t **do** ▷ separator-consistent messages
            $M_c(\sigma, b) \leftarrow \max\{V_c(a_c, b) : a_c \subseteq B_c, \; a_c \cap S_c = \sigma\}$ for each $\sigma \subseteq S_c$, $b \le k$.
        end for
        for each $a_t \subseteq B_t$ and $b \in \{0, \dots, k\}$ **do**
            $q \leftarrow b - |a_t \cap R_t|$.
            if $q < 0$ **then**
                $V_t(a_t, b) \leftarrow -\infty$.
            else
                $V_t(a_t, b) \leftarrow \Phi_t(a_t) + \big(\bigoplus_{c \in \text{ch}(t)} M_c(a_t \cap S_c, \cdot)\big)(q)$. ▷ $\oplus$ = max-plus conv.
            end if
        end for
    end if
end for
return $\max_{a_r \subseteq B_r, \, b \le k} \; V_r(a_r, b)$.

where $\mathcal{F}_s$ denotes the factors assigned to bag s (both top-layer factors f_u and unary tail factors g_i), and the maximum is $-\infty$ if no feasible A exists. Since $\sum_h h(A \cap E(h)) = \sum_u f_u(A \cap E(u)) + \langle m_+, \mathbf{1}_A \rangle = F(A)$ over the augmented family, the root table returns the full objective, tail included.

If t is a leaf, then $T_t = \{t\}$, $V_t^* = R_t$, and the only variables are those in B_t. The assignment a_t fixes which variables are selected, so $|A \cap V_t^*| = |a_t \cap R_t|$. The algorithm sets $V_t(a_t, b) = \Phi_t(a_t)$ when $b = |a_t \cap R_t|$ and $-\infty$ otherwise, matching the claim.

Suppose t has children $c_1, \dots, c_d$ and the claim holds for each child. Consider an assignment $a_t \subseteq B_t$ and budget b. Any feasible set A for the subtree T_t decomposes as: (i) the local assignment $a_t \cap R_t$ contributing $|a_t \cap R_t|$ to the budget and value $\Phi_t(a_t)$ from the local factors, and (ii) for each child c, a selection within the subtree T_c that agrees with a_t on the separator $S_c = B_c \cap B_t$ and uses some budget b_c from the private variables V_c^*.

By the running-intersection property, the private variable sets $V_{c_1}^*, \dots, V_{c_d}^*$ are pairwise disjoint and disjoint from R_t. Therefore the total budget is $b = |a_t \cap R_t| + \sum_{j=1}^d b_{c_j}$, and the total value decomposes as

$$\Phi_t(a_t) + \sum_{j=1}^{d} M_{c_j}(a_t \cap S_{c_j}, \, b_{c_j}),$$

where $M_{c_j}(\sigma, b_{c_j}) = \max\{V_{c_j}(a_{c_j}, b_{c_j}) : a_{c_j} \cap S_{c_j} = \sigma\}$ is the message from child c_j. By the inductive hypothesis, V_{c_j} correctly represents the subtree optimum, so M_{c_j} correctly represents the best child-subtree value consistent with the separator assignment and budget.

The remaining budget $q = b - |a_t \cap R_t|$ is distributed among children by maximizing over all splits $(b_{c_1}, \ldots, b_{c_d})$ with $\sum_j b_{c_j} = q$. This is exactly the iterated max-plus convolution $\big(\bigoplus_{c \in \mathrm{ch}(t)} M_c(a_t \cap S_c, \cdot)\big)(q)$ computed in the convolution step of Algorithm 2.

At the root r, $S_r = \emptyset$ and V_r^* accounts for all variables in V (since every variable is private to exactly one subtree). The final answer $\max_{a_r, b \le k} V_r(a_r, b)$ thus equals $\max_{|A| \le k} F(A)$. □

The modular tail is included by assigning each unary factor $g_i(A) = m_+(i)\mathbf{1}[i \in A]$ to one bag containing i. Separator consistency ensures that its contribution is counted exactly once. Since the supplied decomposition covers V, every unary factor can be assigned to an existing bag without increasing its size.

H.2 Complexity analysis

Proof of Theorem 4.1, complexity. Each bag B_t has $|B_t| \le w + 1$ vertices, so the number of assignments $a_t \subseteq B_t$ is at most 2^{w+1}. The budget dimension has $k + 1$ values.

Let $s_u = |E(u)|$ and let c_u be the arithmetic cost of one evaluation of the sub-DAG producing z_u, including its input values on $E(u)$. Evaluating each top-layer factor independently on all subsets gives

$$T_{\text{tables}} = O\left(\sum_{u \in U_K} 2^{s_u}(1 + c_u) \right). \tag{10}$$

This bound allows repeated evaluation of shared subcomputations. Given the factor-to-bag assignments, initialize each bag potential and add its assigned factors using unit-cost table access. With $I_+ = \{i : m_+(i) > 0\}$, the loading cost, including unary tail factors, is

$$T_{\text{load}} = O\big(2^{w+1}\big(|U_K| + |I_+| + |\mathcal{T}|\big)\big). \tag{11}$$

Subset projections for table lookup can be precomputed in $O(2^{|B_t|})$ operations per assigned factor by subset recurrence. For depth-1 factors, the weighted subset sums can likewise be generated with one addition per nonempty subset, followed by one activation evaluation. Thus their table construction takes $O(\sum_{u \in U_1} 2^{s_u})$ operations once the support weights are supplied.

For each child c and separator assignment $\sigma \subseteq S_c$, the message $M_c(\sigma, b)$ is obtained by maximizing $V_c(a_c, b)$ over all a_c compatible with σ. Since $|S_c| \le w + 1$, there are at most 2^{w+1} separator assignments, and each message entry is a maximum over at most 2^{w+1} table entries per budget value. This contributes $O(2^{2(w+1)} \cdot k)$ per child, but can be reduced to $O(2^{w+1} \cdot k)$ by iterating over all a_c once and updating the relevant $\sigma = a_c \cap S_c$ entry.

At an internal bag t with d children, for each of the 2^{w+1} assignments a_t, the iterated max-plus convolution of d child message tables (each of length $k+1$) is computed by $d-1$ pairwise convolutions. Each pairwise generic max-plus convolution costs $O(k^2)$. Summing over all bags, the convolution cost is $O(|\mathcal{T}| \cdot 2^{w+1} \cdot k^2)$, which bounds the DP updates.

The total cost is $T_{\text{tables}} + T_{\text{load}} + O(|\mathcal{T}|\, 2^{w+1} k^2)$. After the aggregated bag potentials Φ_t are supplied, the DP phase alone is $O(|\mathcal{T}|\, 2^{w+1} k^2)$. Its tables occupy $O(|\mathcal{T}|\, 2^{w+1} k)$ memory. Storing one maximizing choice per table entry permits standard traceback of an optimal set within the same bounds. □

H.3 Tree decomposition computation

For fixed width w, a tree decomposition of width w (if one exists) can be found in $O(n)$ time by the algorithm of Bodlaender (1993), where $n = |V|$. Heuristic elimination orderings may also supply a decomposition, with the DP bound determined by the width returned. In our experiments (Section O), tree decompositions are supplied as part of the instance specification. Automatic decomposition discovery is orthogonal to the DP correctness result.

I Width Lower Bounds and Support Families

Proof of Theorem 4.2. For a graph $G = (V, E)$, create one factor $\min(\mathbf{1}[u \in A] + \mathbf{1}[v \in A], 1)$ for each edge uv. This gives a depth-1 DSF with unit weights, zero modular tail, and effective primal graph exactly G. Its value counts edges incident to A, so

$$\max_{|A| \le k} F_G(A) = |E| \quad \Longleftrightarrow \quad G \text{ has a vertex cover of size at most } k.$$

ETH asserts that 3-SAT has no $2^{o(N)}$ algorithm in the number of variables. By sparsification, it also rules out such an algorithm for sparse 3-SAT (Impagliazzo et al., 1998). Recall the linear-size reduction from a formula with N variables and $M = O(N)$ three-literal clauses. Create an adjacent pair of literal vertices for each variable, a triangle for each clause, and an edge from each clause vertex to its corresponding literal vertex. Set $k = N + 2M$. Every cover takes at least one vertex from each variable pair and two from each clause triangle. A cover of size at most k takes exactly these numbers. Declare the literal whose variable vertex is covered to be true. The uncovered vertex of each clause triangle has its literal vertex covered, so each clause is satisfied. Conversely, a satisfying assignment gives a size-k cover by taking the true variable vertices and leaving one true literal vertex uncovered in each clause triangle.

The graph has $n = 2N + 3M = O(N)$ vertices and $O(N)$ edges. Its DSF encoding has length polynomial in N, and its effective treewidth is at most $n - 1$. A $2^{o(w)}\,\mathrm{poly}(L)$ algorithm would therefore solve these sparse formulas in $2^{o(N)}$ time, contradicting ETH. For the supplied-decomposition formulation, provide the single bag V of width $n - 1$. The same contradiction follows. The reduction establishes the necessary exponential order in width, without fixing its base or the polynomial factors. □

Theorem I.1 (Coverage hardness)**.** *There is a family of depth-*1 *coverage DSFs with unbounded effective treewidth on which no polynomial-time algorithm attains approximation ratio* $1 - 1/e + \varepsilon$ *for any fixed* $\varepsilon > 0$*, unless* $\mathrm{P} = \mathrm{NP}$.

Proof. For Max-k-Cover, let V index available sets and $N(j) \subseteq V$ index the sets containing universe element j. Then $F(A) = \sum_j \min(|A \cap N(j)|, 1)$ is a polynomial-size depth-1 DSF with exactly the coverage objective. Apply Feige's Max-k-Cover hardness (Feige, 1998). This family of encodings has unbounded width: an element incident to all n sets induces a clique of size n. A uniformly bounded-width restriction is exactly solvable by Theorem 4.1 using a fixed-width decomposition (Bodlaender, 1993). □

Corollary I.2 (Fixed minor-closed support families)**.** *Let* $\mathcal{G}$ *be a fixed minor-closed graph family. If its treewidth is bounded, cardinality-constrained DSF maximization on representations with effective primal graph in* $\mathcal{G}$ *is polynomial-time solvable, assuming polynomial-time exact rational local evaluation. If its treewidth is unbounded, the problem is NP-hard already for depth-*1 *unit-cap pairwise factors. On this finite-input subclass its threshold decision version is NP-complete.*

Proof. For bounded width, compute a fixed-width decomposition (Bodlaender, 1993). Top-layer scopes have bounded size, so Boolean factor tables and bag potentials can be built in polynomial time. Apply Theorem 4.1. If $\mathcal{G}$ has unbounded treewidth, it contains every planar graph: excluding a planar minor forces bounded treewidth (Robertson & Seymour, 1986). Vertex Cover is NP-complete on planar graphs (Garey et al., 1976). The edge-coverage construction preserves the graph and reduces Vertex Cover to threshold $|E|$. A selected set is a polynomially verifiable certificate for this unit-cap threshold problem, establishing NP-completeness on the stated subclass. □

This is a worst-case classification of graph families, compatible with tractable objectives of large width such as Example 4.5. These graph minors concern representation supports, whereas Corollary 3.6 concerns matroid minors of target rank functions.

J Budget-Splitting DP: Correctness and Complexity

Proposition J.1 (Budget-splitting DP for tree-structured DSFs)**.** *For depth-*K *computation-tree DSFs with pairwise disjoint child effective supports and no external modular tail (*$m_+ = 0$*), the*

cardinality-constrained maximum is computed exactly in $O(P_{\mathrm{leaf}} + |\mathcal{L}|\, d\, k^2)$ time. Here $|\mathcal{L}|$ counts computation-tree nodes, $d \geq 1$ bounds the arity, and $P_{\mathrm{leaf}} = \sum_\ell |E(\ell)| \log(1 + |E(\ell)|)$ accounts for sorting the leaf-ridge weights. This does not require bounded primal-graph treewidth. The proof is in Section J.

We give a formal correctness proof and complexity analysis for the budget-splitting DP of Proposition J.1.

Proof of Proposition J.1. Represent the linear output by an identity root r, with the output coefficients as its incoming weights. Remove zero-weight edges and recursively prune identically zero non-root nodes and components that do not contribute to r. If r has no children after pruning, then $F \equiv 0$: set $V_r(b) = 0$ for every $b = 0, \ldots, k$ and return the empty set with objective value zero. Otherwise, every non-root computation node then feeds exactly one parent. Let $\mathcal{L}$ denote the resulting nodes and d their maximum arity, including the output root. This root does not increase the nonlinear depth. For the subfunction F_u rooted at u, define

$$V_u(b) = \max\{F_u(A) : A \subseteq E(u),\ |A| \leq b\}, \qquad b = 0, \ldots, k.$$

These tables are non-decreasing and finite, including when $b > |E(u)|$. At a layer-1 node, let $w_{u,\pi(1)} \geq \cdots \geq w_{u,\pi(|E(u)|)}$ be its sorted nonnegative weights. Monotonicity of φ_u gives

$$V_u(b) = \varphi_u\left(\sum_{j=1}^{\min(b,|E(u)|)} w_{u,\pi(j)}\right).$$

Suppose the child tables at an internal node u are correct. Its pairwise disjoint child supports give the unique decomposition $A = \bigsqcup_{c \in \mathrm{ch}(u)} (A \cap E(c))$. Consequently,

$$W_u(b) = \max_{\substack{\sum_c b_c = b \\ b_c \geq 0}} \sum_{c \in \mathrm{ch}(u)} w_{uc} V_c(b_c), \qquad V_u(b) = \varphi_u(W_u(b)).$$

To verify the recurrence, any feasible A induces child cardinalities whose sum is at most b. Increasing the allocated budgets until their sum is b preserves feasibility in the at-most tables and bounds the pre-activation of A by $W_u(b)$. Conversely, maximizing sets for any allocation $(b_c)_c$ have a disjoint union of cardinality at most b and attain the corresponding pre-activation. Monotonicity of φ_u then proves the displayed identity. Induction yields $V_r(k) = \max_{|A| \leq k} F(A)$.

Sorting the leaf weights costs $O(P_{\mathrm{leaf}})$, where $P_{\mathrm{leaf}} = \sum_{\ell\ \mathrm{leaf}} |E(\ell)| \log(1 + |E(\ell)|)$. Filling each leaf table costs $O(k)$. At an internal node, scaling the child tables and applying the activation cost $O((d_u + 1)k)$, and iterated max-plus convolution costs $O(d_u k^2)$. Thus the total cost is $O(P_{\mathrm{leaf}} + |\mathcal{L}| d k^2)$, with d taken to be at least one. □

A nonnegative sum of disjoint rooted components can be handled by adding an identity root. The proposition assumes no external modular tail, since adding arbitrary element scores need not preserve the leaf selection rule or the discrete concavity used by the acceleration.

J.1 Aligned laminar-matroid constraints

Consider the disjoint-support computation trees without an external modular tail in Proposition J.1. Assign integer caps $c_u \geq 0$ to a subfamily of node supports, requiring $|A \cap E(u)| \leq c_u$, and set $c_u = k$ at unconstrained nodes. These supports are laminar, so the constraints define a laminar matroid. At a leaf, truncate the sorted-weight table at c_u. At an internal node, use the already constrained child tables to form

$$W_u^{\mathrm{cap}}(b) = \max_{\substack{\sum_c b_c = b \\ b_c \geq 0}} \sum_{c \in \mathrm{ch}(u)} w_{uc} V_c^{\mathrm{cap}}(b_c), \qquad V_u^{\mathrm{cap}}(b) = \varphi_u(W_u^{\mathrm{cap}}(\min(b, c_u))).$$

Disjointness makes child feasibility independent, and induction proves that this table optimizes over all feasible sets of cardinality at most b. Under an exact-budget convention, the equivalent operation assigns $-\infty$ to entries with $b > c_u$. Truncating an at-most table replaces a suffix of its nonnegative,

non-increasing marginal sequence by zeros, preserving discrete concavity. Thus the generic and accelerated runtimes remain $O(P_{\text{leaf}} + |\mathcal{L}|dk^2)$ and $O(P_{\text{leaf}} + |\mathcal{L}|dk)$, respectively.

A root-only cap defines a uniform matroid on $E(r)$, with elements of $V \setminus E(r)$ free. When $E(r) = V$, this is a uniform matroid on V. Caps on disjoint represented supports give aligned partition constraints.

K Budget-Table Concavity

K.1 Concave composition lemma

Lemma K.1 (Concave composition preserves discrete concavity)**.** *Let $V : \{0, 1, \dots, k\} \to \mathbb{R}_+$ be non-decreasing and discretely concave (i.e. $\Delta V(b) := V(b) - V(b-1)$ is non-increasing in b). Let $\varphi : \mathbb{R}_+ \to \mathbb{R}_+$ be concave and non-decreasing. Then $\varphi \circ V$ is non-decreasing and discretely concave.*

Proof. Monotonicity follows by composition. For $1 \le b < k$, discrete concavity of V gives $V(b) \ge (V(b-1) + V(b+1))/2$. Hence

$$\varphi(V(b)) \ge \varphi\left(\frac{V(b-1) + V(b+1)}{2}\right) \ge \frac{\varphi(V(b-1)) + \varphi(V(b+1))}{2},$$

by monotonicity and concavity of φ, respectively. This is discrete concavity of $\varphi \circ V$. □

K.2 Budget-table concavity for tree-structured DSFs

Lemma K.2 (Budget-table concavity for tree-structured DSFs)**.** *In the laminar/tree-structured budget-splitting DP of Proposition J.1, with concave non-decreasing activations satisfying $\varphi_u(0) = 0$ for every nonlinear node u, the value table $V_t(b)$ at each node is non-decreasing and discretely concave in the allocated cardinality budget b.*

Proof of Lemma K.2. We use Theorem L.1, proved independently in Section L, at every internal merge step. The proof is by induction over the computation tree, processed bottom-up.

A layer-1 node t has modular weights $w_1 \ge w_2 \ge \cdots \ge w_{n_t} \ge 0$ (sorted in non-increasing order) and concave non-decreasing activation φ_t with $\varphi_t(0) = 0$. The pre-activation prefix table

$$P_t(b) \;:=\; \sum_{i=1}^{\min(b,\, n_t)} w_i, \qquad b = 0, 1, \dots, k,$$

has non-increasing marginal gains $\Delta P_t(b) = w_b$ for $b \le n_t$ and 0 for $b > n_t$, so P_t is non-decreasing and discretely concave with $P_t(0) = 0$. Applying the activation gives the value table $V_t(b) = \varphi_t(P_t(b))$. Since φ_t is concave non-decreasing and P_t is non-decreasing and discretely concave, Lemma K.1 gives that V_t is non-decreasing and discretely concave, with $V_t(0) = \varphi_t(0) = 0$.

Let t be an internal node with children $c_1, \dots, c_d$ (with disjoint effective supports), inter-layer weights $w_{t,c_j} \ge 0$, and activation φ_t that is concave and non-decreasing. By the inductive hypothesis, each child table $V_{c_j}(\cdot)$ is non-decreasing and discretely concave, with $V_{c_j}(0) = 0$.

Define $\widehat{V}_{c_j}(b) := w_{t,c_j}\, V_{c_j}(b)$. Since $w_{t,c_j} \ge 0$, each $\widehat{V}_{c_j}$ is non-decreasing and discretely concave, with $\widehat{V}_{c_j}(0) = 0$.

The combined pre-activation table is obtained by iterated max-plus convolution of the weighted tables:

$$W_t(b) \;:=\; \big(\bigoplus_{j=1}^{d} \widehat{V}_{c_j}\big)(b) \;=\; \max_{\substack{b_1 + \cdots + b_d = b \\ b_j \ge 0}} \sum_{j=1}^{d} w_{t,c_j}\, V_{c_j}(b_j).$$

By Theorem L.1 (applied iteratively), W_t is non-decreasing and discretely concave, with $W_t(0) = 0$.

The value table at node t is $V_t(b) = \varphi_t(W_t(b))$. Since W_t is non-decreasing and discretely concave and φ_t is concave non-decreasing (hypothesis), Lemma K.1 gives that V_t is non-decreasing and discretely concave.

At $b = 0$: $V_t(0) = \varphi_t(W_t(0)) = \varphi_t(0)$. For normalized activations with $\varphi_t(0) = 0$, $V_t(0) = 0$. The composition lemma applies starting from $b = 1$ with $a = W_t(0) = 0$, $c = W_t(1)$, $d = W_t(2)$. □

K.3 Greedy exactness on disjoint computation trees

Proof of Corollary 4.4. For a monotone function h on a ground set E, write $V_h(b) = \max_{A \subseteq E, |A|=b} h(A)$ for $0 \le b \le |E|$. We prove jointly that every greedy prefix attains $V_h(b)$ and that V_h is nondecreasing and discretely concave. Both statements hold for nonnegative modular functions by sorting their weights.

Suppose they hold for functions h_c on disjoint sets E_c, and let $H(A) = \sum_c w_c h_c(A \cap E_c)$ with $w_c \ge 0$. For $w_c > 0$, the subsequence selected in E_c by greedy for H is greedy for h_c: a larger available marginal in the same component would contradict the global greedy choice. After b_c selections, the next available weighted marginal is therefore

$$w_c\big(V_{h_c}(b_c + 1) - V_{h_c}(b_c)\big).$$

These sequences are nonnegative and non-increasing. Greedy merges them by repeatedly taking a largest available entry. Every allocation of b elements chooses a prefix from each sequence, so the merged prefix is optimal. Its increments are non-increasing. Components with $w_c = 0$ contribute zero marginal sequences and satisfy the same conclusion.

Next let $f = \varphi \circ H$, where H has both induction properties and φ is concave and nondecreasing. Monotonicity gives $V_f(b) = \varphi(V_H(b))$, which is discretely concave by Lemma K.1. Consider any greedy sequence for f. If it is also greedy for H, all its prefixes are optimal for both functions. Otherwise consider its first choice that is not greedy for H, from a prefix A. Put $h = H(A)$, and let a be the selected H-marginal and b the largest available H-marginal, so $0 \le a < b$. The transformed gains tie, giving $\varphi(h + a) = \varphi(h + b)$. A concave nondecreasing function with $\varphi(x) = \varphi(y)$ at $x < y$ is constant on $[x, \infty)$, since for $z > y$,

$$0 \le \frac{\varphi(z) - \varphi(y)}{z - y} \le \frac{\varphi(y) - \varphi(x)}{y - x} = 0.$$

Thus the selected prefix attains the global maximum of f, as do all later prefixes. Earlier prefixes are H-greedy and hence f-optimal. This proves optimality for every tie-breaking rule.

Induction through disjoint weighted sums and concave activations proves both properties at the identity output root. Elements outside its support are zero modular components. Finally, monotonicity equates the exact-cardinality and at-most-cardinality optima for $0 \le k \le |V|$. □

K.4 Separator-indexed budget tables: definition and a non-concavity counterexample

Definition K.3 (Separator-indexed budget table)**.** In the junction-tree DP of Theorem 4.1, after marginalising over subtree variables below separator S, the *separator-indexed budget table* $M(\sigma, b)$ records, for each separator assignment $\sigma \subseteq S$ and cumulative budget $b \in \{0, \dots, k\}$ used in the subtree, the best partial objective value:

$$M(\sigma, b) \;:=\; \max\Big\{ \sum_{u \in \mathcal{F}_{\mathrm{sub}}} f_u(A \cap E(u)) \;:\; A \subseteq V^*_{\mathrm{sub}} \cup S,\; A \cap S = \sigma,\; |A \cap V^*_{\mathrm{sub}}| = b \Big\}.$$

In Lemma K.2, disjoint child supports make the pre-activation table a max-plus convolution of independent child tables. The following four-element example isolates the role of this independence: overlapping factors yield a non-concave budget table even at bounded width.

Proposition K.4 (Generic separator concavity fails)**.** *Separator-indexed budget tables $M(\sigma, \cdot)$ need* not *be discretely concave for general bounded-treewidth depth-1 DSFs. There is a depth-1 SCMM on $V = \{1, 2, 3, 4\}$ whose exact-cardinality table $M(b) = \max_{|A|=b} F(A)$ (equivalently the separator table with $S = \varnothing$) equals $(0, 7, 10, 14, 14)$, with marginal sequence $(7, 3, 4, 0)$. Since $4 > 3$ the marginals are not non-increasing.*

Proof. Let $F = \sum_{u=1}^{4} f_u$, where $f_u(A) = \min(\langle w_u, \mathbf{1}_A \rangle, c_u)$ and

u	w_{u1}	w_{u2}	w_{u3}	w_{u4}	c_u
1	0	0	2	2	2
2	0	2	2	0	4
3	3	0	1	2	4
4	1	3	0	3	4

All weights and caps are nonnegative, so F is a depth-1 SCMM. The primal graph is K_4 (treewidth 3, bounded). Direct enumeration gives the cardinality-optimal sets $\varnothing, \{4\}, \{1,4\}, \{1,2,3\}, \{1,2,3,4\}$ with values $0, 7, 10, 14, 14$: e.g. $F(\{4\}) = 2+0+2+3 = 7$, every pair attains at most 10 (e.g. $F(\{1,4\}) = 2+0+4+4 = 10$), and $F(\{1,2,3\}) = 2+4+4+4 = 14$. The marginal from $b = 2$ to $b = 3$ is 4, exceeding the marginal 3 from $b = 1$ to $b = 2$, so M is not discretely concave. The same profile is realized as a genuine child message $M_c(\varnothing, \cdot)$ by attaching the bag $\{1,2,3,4\}$ to a parent bag with empty intersection (so $S = \varnothing$). (Both the exact-b and at-most-b tables coincide here and are non-concave.) □

Thus bounded effective treewidth alone does not imply budget-table concavity. The generic junction-tree DP remains exact, while its linear-time convolution acceleration requires the additional concavity hypothesis.

For a fixed separator assignment, subtract the constant $M(\sigma, 0)$ before marginal merging and add it back afterward. Under exact-budget semantics, merge only finite feasible-prefix marginals and retain $-\infty$ outside the attainable range. Infeasible cardinalities remain assigned $-\infty$.

Corollary K.5 (Conditional concavity-accelerated JT-DP)**.** *Under the hypotheses of Theorem 4.1, if all separator-indexed budget tables in the junction-tree DP are discretely concave, then the DP phase improves to $O(|\mathcal{T}| \cdot 2^{w+1} \cdot k)$. The total cost additionally includes $T_{\text{tables}} + T_{\text{load}}$ from* (10)–(11)*.*

L Concave Max-Plus Convolution

The marginal merge is the standard slope merge for concave polygonal chains, equivalently the upper-boundary operation in the linear-time Minkowski-sum algorithm for convex polygons (de Berg et al., 2008, cf. Theorem 13.10).

Theorem L.1 (Concavity-accelerated max-plus convolution)**.** *Let $a, b : \{0, 1, \ldots, k\} \to \mathbb{R}$ be non-decreasing and discretely concave, with $a(0) = b(0) = 0$. Their max-plus convolution*

$$c(n) \;:=\; \max_{0 \le i \le n} \big(a(i) + b(n-i)\big), \qquad n = 0, \ldots, k,$$

can be computed in $O(k)$ time by marginal-increment merging. Moreover, c is itself non-decreasing and discretely concave.

Proof. Let $\Delta a(i) = a(i) - a(i-1)$ and $\Delta b(i) = b(i) - b(i-1)$ for $i = 1, \ldots, k$. Both sequences are nonnegative and non-increasing. Merge them in non-increasing order, preserving the order within each sequence when values tie, and denote the result by $\delta_1 \ge \cdots \ge \delta_{2k}$. For every $0 \le i \le n$,

$$a(i) + b(n-i) = \sum_{r=1}^{i} \Delta a(r) + \sum_{r=1}^{n-i} \Delta b(r) \le \sum_{r=1}^{n} \delta_r.$$

The first n merged entries consist of a prefix of length i from Δa and a prefix of length $n-i$ from Δb, for some i. This split attains equality, so $c(n) = \sum_{r=1}^{n} \delta_r$. The merge and prefix sums take $O(k)$ time. Since $\Delta c(n) = \delta_n \ge 0$ and $\delta_n \ge \delta_{n+1}$, c is non-decreasing and discretely concave. □

M Controlled Binary Universality

A *rank-one SCMM atom* is $A \mapsto \kappa \min(\langle m, \mathbf{1}_A \rangle, \theta)$ or a modular $A \mapsto \kappa \langle m, \mathbf{1}_A \rangle$ with $\kappa, \theta \ge 0$ and $m \ge 0$. A *controlled expression* is a finite tree over such atoms using nonnegative sums and inf-convolution $(g \square h)(A) := \min_{B \subseteq A} \big(g(B) + h(A \setminus B)\big)$ in which *every subexpression is itself in* $\mathrm{MonSub}(V)$. Write C-SCMM(V) for the functions expressible using only these shallow atoms.

By Definition 5.1, C-SCMM$(V) \subseteq$ CDSF$(V) \subseteq$ MonSub(V). Equality will follow from the construction.

We prove Theorem 5.2 of Section 5: C-SCMM$(V) =$ CDSF$(V) =$ MonSub(V). The proof is a deletion–contraction induction on $|V|$. Fix $f \in \mathrm{MonSub}(V)$, a pivot $e \in V$, and set $W = V \setminus \{e\}$, $\alpha = f(\{e\})$.

M.1 Failure of submodularity under inf-convolution

For monotone g, h, their convolution is monotone: if $A \subseteq B$, any split $D \subseteq B$ induces the split $D \cap A$ of A, with each summand no larger. Minimizing proves $(g \square h)(A) \leq (g \square h)(B)$. Submodularity can fail. On $V = \{0, 1, 2\}$, take normalized monotone submodular functions with values

A	0	1	2	01	02	12	012
$g(A)$	3	1	2	3	3	2	3
$h(A)$	1	1	3	1	3	3	3

(single digits denote singletons). Directly, $q = g \square h$ has $q(01) = 1$, $q(12) = 2$, $q(012) = 3$, $q(1) = 1$, violating submodularity. Each binary output below is instead certified by equality to its target function.

M.2 Auxiliary lemmas

Lemma M.1 (Deletion and normalized contraction)**.** *The deletion $d(S) := f(S)$ and contraction $c(S) := f(S \cup \{e\}) - \alpha$ ($S \subseteq W$) are in* MonSub(W).

Proof. Restriction preserves normalization, monotonicity, and submodularity, so $d \in \mathrm{MonSub}(W)$. The definition of c gives $c(\emptyset) = 0$, and monotonicity of f gives $0 \leq c(S) \leq c(T)$ whenever $S \subseteq T$. For $S, T \subseteq W$, submodularity of f yields

$$\begin{aligned} c(S) + c(T) &= f(S \cup \{e\}) + f(T \cup \{e\}) - 2\alpha \\ &\geq f((S \cup T) \cup \{e\}) + f((S \cap T) \cup \{e\}) - 2\alpha \\ &= c(S \cup T) + c(S \cap T). \end{aligned}$$

Thus $c \in \mathrm{MonSub}(W)$. □

For $p : 2^W \to \mathbb{R}$ let $Lp(A) := p(A \cap W)$.

Lemma M.2 (Extension by a dummy coordinate)**.** *L preserves* MonSub *and maps rank-one SCMM atoms to rank-one SCMM atoms. Moreover $L(g + h) = Lg + Lh$ and $Lg \square_V Lh = L(g \square_W h)$. Consequently a controlled W-expression lifts to a controlled V-expression, every node still in* MonSub(V).

Proof. Intersecting with W commutes with unions and intersections, so L preserves normalization, monotonicity, and submodularity. Extending an atom's weight vector by a zero at e gives its lift, and $L(g + h) = Lg + Lh$ follows pointwise. For any $A \subseteq V$,

$$\begin{aligned} (Lg \square_V Lh)(A) &= \min_{B \subseteq A} \big(g(B \cap W) + h((A \setminus B) \cap W)\big) \\ &= \min_{C \subseteq A \cap W} \big(g(C) + h((A \cap W) \setminus C)\big) \\ &= L(g \square_W h)(A). \end{aligned}$$

The second equality holds because $B \cap W$ ranges over all subsets of $A \cap W$. Applying these identities at every node lifts the expression, and preservation of MonSub certifies every lifted subexpression. □

Lemma M.3 (Convolution of two normalized upper bounds)**.** *Let $q, g, h : 2^V \to \mathbb{R}$ be* normalized *($q(\emptyset) = g(\emptyset) = h(\emptyset) = 0$) with q subadditive, $q \leq g$, $q \leq h$ pointwise, and* $\min(g, h) = q$. *Then $g \square h = q$.*

Proof. The splits $B = A$ and $B = \emptyset$, together with normalization, give

$$(g\square h)(A) \le \min\{g(A), h(A)\} = q(A).$$

Conversely, for every $B \subseteq A$, the pointwise bounds and subadditivity of q give

$$g(B) + h(A \setminus B) \ge q(B) + q(A \setminus B) \ge q(A).$$

Taking the minimum over B proves the reverse inequality. □

M.3 Binary deletion–contraction factorization

Write $\bar{d}(A) = f(A \setminus \{e\})$ and $\bar{c}(A) = f((A \setminus \{e\}) \cup \{e\}) - \alpha$ (the lifts of d, c), and set

$$G_e(A) = \bar{d}(A) + \alpha\,\mathbf{1}[e \in A], \qquad H_e(A) = \bar{c}(A) + \alpha\,\mathbf{1}[A \neq \emptyset].$$

Lemma M.4. *$G_e, H_e \in \mathrm{MonSub}(V)$ are normalized, and $f = G_e \square H_e$.*

Proof. By Lemmas M.1 and M.2, $\bar{d}$ and $\bar{c}$ belong to $\mathrm{MonSub}(V)$. The added functions are rank-one SCMM atoms:

$$\alpha\mathbf{1}[e \in A] = \alpha \min(\langle \mathbf{1}_{\{e\}}, \mathbf{1}_A \rangle, 1), \qquad \alpha\mathbf{1}[A \neq \emptyset] = \alpha \min(\langle \mathbf{1}_V, \mathbf{1}_A \rangle, 1).$$

Therefore $G_e, H_e \in \mathrm{MonSub}(V)$ and both vanish at $\emptyset$. If $e \notin A$, then $G_e(A) = f(A)$ and, for nonempty A, $H_e(A) = f(A \cup \{e\}) \ge f(A)$ by monotonicity. If $e \in A$, then $H_e(A) = f(A)$ and

$$G_e(A) = f(A \setminus \{e\}) + f(\{e\}) \ge f(A)$$

by normalized submodularity. Hence $\min\{G_e, H_e\} = f$ on all subsets. Since f is subadditive, Lemma M.3 gives $G_e \square H_e = f$. □

M.4 Universality induction

Proof of Theorem 5.2. Induct on $|V|$. If $|V| = 0$, $f \equiv 0$ (empty nonnegative sum). If $|V| = 1$, $f = \alpha\,\mathbf{1}[e \in A]$, a single atom. For $|V| \ge 2$ pick a pivot e. By Lemma M.1, $d, c \in \mathrm{MonSub}(W)$, so by the inductive hypothesis they have controlled expressions, which lift to controlled V-expressions (Lemma M.2). Adding the two rank-one atoms forms G_e, H_e as controlled expressions in $\mathrm{MonSub}(V)$, and $f = G_e \square H_e$ (Lemma M.4) is a valid controlled step because its output equals $f \in \mathrm{MonSub}(V)$. Every subexpression lies in $\mathrm{MonSub}(V)$, so $f \in \mathrm{CDSF}(V)$. The leaves are depth-1 SCMM atoms, so $f \in \text{C-SCMM}(V)$. With $\text{C-SCMM}(V) \subseteq \mathrm{CDSF}(V) \subseteq \mathrm{MonSub}(V)$, all three coincide.

Each level spends one element and branches into d and c, so $S(n) \le 2S(n-1) + O(1) = O(2^n)$ expression nodes. With explicit atom supports this uses $O(n\,2^n)$ support/parameter slots, in addition to the encoding length of the real-valued parameters inherited from f. The atom $\alpha \min(\langle \mathbf{1}_V, \mathbf{1}_A \rangle, 1)$ has full support, so the construction imposes no bound on effective treewidth. □

N Depth-K Effective Hypergraph Behavior

For a depth-K DSF, use the effective supports of the original nonlinear nodes from Definition 2.2. For the computation-tree recursion, augment the tree with the identity output root r and set $E(r) = \bigcup_{c: w_{rc} > 0} E(c)$. This auxiliary support is used only in the recursion: the effective clique hypergraph still consists of the supports of the original nonlinear nodes. Adding the identity output root therefore adds no hyperedge and does not change the effective primal graph or its treewidth.

Proposition N.1. *If layer-1 cliques $\{C_j\}$ are laminar and the computation nodes, including the identity output root, form a rooted tree as specified before Proposition J.1, with $\bigcup_{c \in \mathrm{children}(u)} E(c) = \bigsqcup_c E(c)$ (disjoint child unions), then the effective clique hypergraph at every depth is laminar.*

Proof. Consider any two nodes u, v. If one is an ancestor of the other, recursive support inclusion gives $E(u) \subseteq E(v)$ or $E(v) \subseteq E(u)$. Otherwise, their paths to the root first meet at a node w, and u, v lie in distinct child subtrees of w. Their supports are contained in the disjoint supports of those two children. Hence $E(u) \cap E(v) = \emptyset$. Every pair of supports in this augmented tree is therefore nested or disjoint. The supports of the original nonlinear nodes form a subfamily, so the effective clique hypergraph is laminar. □

For fully connected DSFs with positive inter-layer weights, each layer-2 node has effective support equal to the union of the layer-1 supports. Its effective primal graph is complete on that union.

The algorithms in Section 4 exploit these support patterns through bounded-width decompositions of top-layer local factors and budget splitting along disjoint computation branches. Example 4.5 illustrates the latter mechanism at maximal effective width.

N.1 Intrinsic maximal width

Proof of the width claim in Example 4.5. For distinct i, j, the mixed difference $\Delta_{ij}F(\emptyset) = F(\{i,j\}) - F(\{i\}) - F(\{j\})$ is $\sqrt{2} - 2$ across the two blocks and $2^{1/4} - 2$ within a block. Both are nonzero. In any additive factorization, absence of a factor containing both i and j would make this difference zero. Thus every additive factorization has complete primal graph. Applying (4) to any DSF representation gives effective width $n - 1$. □

O Experimental Details and Additional Results

O.1 Experimental setup

The synthetic experiments run on a single CPU core (Intel Xeon server). CIFAR-100 downstream classifiers are trained on NVIDIA A100 GPUs.

Timing protocol. Synthetic timings use wall-clock measurements. Each seed contributes the minimum of five timed calls for JT-DP scaling, tree acceleration, and marginal-merge convolution, or two calls for brute-force convolution. Tables and figures report the mean and standard deviation across seeds. JT-DP exactness timings use one call per solver and instance.

Seed policy. The synthetic experiments use seed indices $0 \ldots 19$ (20 per configuration).

Correctness threshold. The floating-point correctness gate is 10^{-10}: any $|F(A_{\mathrm{DP}}) - F(A_{\mathrm{BF}})| > 10^{-10}$ is flagged as a failure. All reported gaps are orders of magnitude below this threshold.

O.2 JT-DP exactness: full results

We evaluate three depth-1 families with supplied tree decompositions: paths of triangles ($w = 2$), paths of 4-cliques ($w = 3$), and planted adversarial coverage traps ($w = 2$). The grid uses $n \in \{12, 16, 20, 24\}$ and $k \in \{4, 6, 8, 12\}$. Triangle and four-clique scopes advance by two and three vertices, respectively, with a final scope covering the last vertices. Each ordinary factor independently chooses $\sqrt{t}$, $\log(1+t)$, or $\min(t, c)$ uniformly, with integer c uniform from one to its scope size. Across 320 instances per family, JT-DP matches exhaustive search to within 7.1×10^{-15}. Greedy is suboptimal on 29.3% of the full suite, with a worst approximation ratio of 0.765. Table 1 reports the complete per-configuration results.

O.3 Adversarial construction details

The planted adversarial family embeds *coverage-trap blocks* into a path-of-triangles graph (treewidth $w = 2$). Each trap block operates on a triangle (a, b, c) and places:

- m unit-cap factors on $\{a, b\}$ and m on $\{a, c\}$: vertex a participates in $2m$ factors, while b and c each participate in m.
- p unit-cap factors on $\{b\}$ and p on $\{c\}$: these single-vertex factors boost the standalone value of b and c.

All activations are $\min(t, 1)$ (unit cap).

Local two-budget analysis. At budget $k = 1$, the greedy marginals are $F(\{a\}) = 2m$, $F(\{b\}) = m + p$, and $F(\{c\}) = m + p$. Greedy selects a when $2m > m + p$, i.e. $p < m$. At budget $k = 2$, the optimal pair is $\{b, c\}$ with value $2m + 2p$ (every factor is covered), versus $\{a, b\}$ or $\{a, c\}$ with

Table 1: Full per-configuration JT-DP results. Each row averages 20 seeds. OPT is brute-force enumeration. Greedy/OPT reports mean across seeds (parenthesized: worst single seed).

family	w	n	k	OPT (BF)	JT-DP	greedy	greedy/OPT (worst)
path-of-triangles (w=2)	2	12	4	7.033	7.033	6.980	0.992 (0.940)
path-of-triangles (w=2)	2	12	6	8.440	8.440	8.413	0.997 (0.956)
path-of-triangles (w=2)	2	12	8	9.320	9.320	9.304	0.998 (0.962)
path-of-triangles (w=2)	2	12	12	10.362	10.362	10.362	1.000 (1.000)
path-of-triangles (w=2)	2	16	4	7.233	7.233	7.175	0.992 (0.917)
path-of-triangles (w=2)	2	16	6	10.086	10.086	10.078	0.999 (0.981)
path-of-triangles (w=2)	2	16	8	11.525	11.525	11.504	0.998 (0.962)
path-of-triangles (w=2)	2	16	12	13.047	13.047	13.047	1.000 (1.000)
path-of-triangles (w=2)	2	20	4	7.693	7.693	7.693	1.000 (1.000)
path-of-triangles (w=2)	2	20	6	10.514	10.514	10.400	0.989 (0.940)
path-of-triangles (w=2)	2	20	8	13.059	13.059	13.058	1.000 (0.999)
path-of-triangles (w=2)	2	20	12	15.514	15.514	15.513	1.000 (0.998)
path-of-triangles (w=2)	2	24	4	7.679	7.679	7.679	1.000 (1.000)
path-of-triangles (w=2)	2	24	6	11.248	11.248	11.233	0.999 (0.972)
path-of-triangles (w=2)	2	24	8	14.000	14.000	13.961	0.997 (0.969)
path-of-triangles (w=2)	2	24	12	16.936	16.936	16.931	1.000 (0.995)
path-of-4-cliques (w=3)	3	12	4	5.919	5.919	5.919	1.000 (1.000)
path-of-4-cliques (w=3)	3	12	6	6.833	6.833	6.833	1.000 (1.000)
path-of-4-cliques (w=3)	3	12	8	7.562	7.562	7.562	1.000 (1.000)
path-of-4-cliques (w=3)	3	12	12	7.995	7.995	7.995	1.000 (1.000)
path-of-4-cliques (w=3)	3	16	4	6.127	6.127	6.122	0.999 (0.983)
path-of-4-cliques (w=3)	3	16	6	8.063	8.063	8.063	1.000 (1.000)
path-of-4-cliques (w=3)	3	16	8	7.851	7.851	7.851	1.000 (1.000)
path-of-4-cliques (w=3)	3	16	12	9.766	9.766	9.766	1.000 (1.000)
path-of-4-cliques (w=3)	3	20	4	7.303	7.303	7.275	0.996 (0.961)
path-of-4-cliques (w=3)	3	20	6	9.140	9.140	9.090	0.995 (0.940)
path-of-4-cliques (w=3)	3	20	8	11.260	11.260	11.251	0.999 (0.983)
path-of-4-cliques (w=3)	3	20	12	13.079	13.079	13.079	1.000 (1.000)
path-of-4-cliques (w=3)	3	24	4	7.250	7.250	7.166	0.988 (0.959)
path-of-4-cliques (w=3)	3	24	6	9.633	9.633	9.605	0.997 (0.973)
path-of-4-cliques (w=3)	3	24	8	11.737	11.737	11.728	0.999 (0.985)
path-of-4-cliques (w=3)	3	24	12	13.868	13.868	13.868	1.000 (1.000)
planted adv. (w=2)	2	12	4	115.415	115.415	100.595	0.875 (0.768)
planted adv. (w=2)	2	12	6	173.173	173.173	162.089	0.943 (0.782)
planted adv. (w=2)	2	12	8	228.992	228.992	222.683	0.975 (0.885)
planted adv. (w=2)	2	12	12	307.800	307.800	307.800	1.000 (1.000)
planted adv. (w=2)	2	16	4	112.269	112.269	95.741	0.850 (0.775)
planted adv. (w=2)	2	16	6	162.416	162.416	144.903	0.889 (0.765)
planted adv. (w=2)	2	16	8	228.294	228.294	216.243	0.948 (0.881)
planted adv. (w=2)	2	16	12	326.693	326.693	321.755	0.985 (0.932)
planted adv. (w=2)	2	20	4	106.791	106.791	89.564	0.840 (0.775)
planted adv. (w=2)	2	20	6	171.676	171.676	145.051	0.845 (0.770)
planted adv. (w=2)	2	20	8	226.841	226.841	202.197	0.893 (0.772)
planted adv. (w=2)	2	20	12	346.120	346.120	333.510	0.965 (0.894)
planted adv. (w=2)	2	24	4	112.079	112.079	97.585	0.873 (0.768)
planted adv. (w=2)	2	24	6	171.719	171.719	151.059	0.879 (0.777)
planted adv. (w=2)	2	24	8	242.284	242.284	214.370	0.884 (0.767)
planted adv. (w=2)	2	24	12	344.489	344.489	324.857	0.942 (0.850)

value $2m + p$. The greedy pair $\{a, b\}$ achieves ratio $(2m + p)/(2m + 2p)$. The constraint $p > \frac{3}{7}m$ ensures this ratio is below 0.85. Setting $m = 15$, $p = 10$ gives greedy/OPT $= 40/50 = 0.800$.

For each full instance, m is uniform on $\{12, 15, 18, 20\}$ and p is uniform on the integers from $\lceil 3m/7 \rceil + 1$ to $m - 1$. We uniformly choose $\min(B, \lceil k/2 \rceil)$ of the B triangle scopes for traps and use random-activation factors on the remaining scopes. Brute-force enumeration evaluates the resulting global greedy gap.

O.4 JT-DP RUNTIME SCALING

To assess the empirical runtime dependence on the parameters k, n, and w, we run three targeted sweeps with 20 seeds each (Figure 3):

- **Vary** k: $n = 100$, $w = 2$ (path-of-triangles), $k \in \{10, 20, 40, 60, 80, 100\}$.
- **Vary** w: $n = 60$, $k = 20$, $w \in \{2, 3\}$ (triangles vs. 4-cliques).
- **Vary** n: $w = 2$, $k = 20$, $n \in \{20, 40, 60, 80, 100\}$.

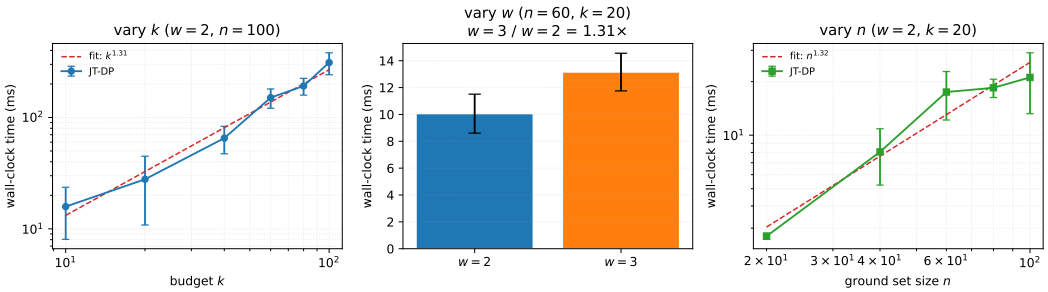


Figure 3: JT-DP wall-clock time vs. k, w, and n. Left: log-log time vs. k with fitted slope. Center: bar chart $w = 2$ vs. $w = 3$. Right: log-log time vs. n with fitted slope. All points are mean $\pm$ std over 20 seeds.

The measured scaling is broadly consistent with the complexity bound $O(|\mathcal{T}| \cdot 2^{w+1} \cdot k^2)$: runtime grows superlinearly in k and roughly linearly in n (which determines $|\mathcal{T}|$). The observed $w = 3$ to $w = 2$ runtime ratio is $1.3\times$. These two sweeps use different factor families, and the asymptotic bound does not specify their finite runtime ratio.

O.5 GREEDY SUBOPTIMALITY DISTRIBUTION

Figure 4 shows the distribution of greedy/OPT ratios across all 960 JT-DP instances, separated by family. The planted adversarial family exhibits a broad distribution of ratios (0.765–1.0), confirming that the coverage-trap construction reliably produces greedy-suboptimal instances. The path-of-triangles and path-of-4-cliques families are predominantly greedy-optimal, with occasional suboptimal instances at small budgets.

O.6 TREE-STRUCTURED ACCELERATION: FULL RESULTS

We compare generic and concavity-accelerated budget-splitting DP on random computation trees with pairwise disjoint child supports and no external modular tail. The grid comprises arities $d \in \{2, 3, 4\}$, ground-set sizes $n \in \{50, 100, 200, 500\}$, and budgets $k \in \{20, 50, 100\}$, with 20 seeds per configuration (720 instances). The accelerated DP achieves a $24.3\times$ speedup at $(n, k, d) = (500, 100, 2)$. This experiment evaluates the complete DP of Corollary 4.3.

Table 2 reports the $d = 2$ summary referenced in the main text. Figure 5 shows the full three-panel grid across $d \in \{2, 3, 4\}$, with log-log wall-clock time vs. k for both the generic and accelerated DPs. The speedup pattern is consistent across arities: the accelerated method grows much more slowly with k than the generic method, consistent with replacing the quadratic budget convolution by marginal merge.

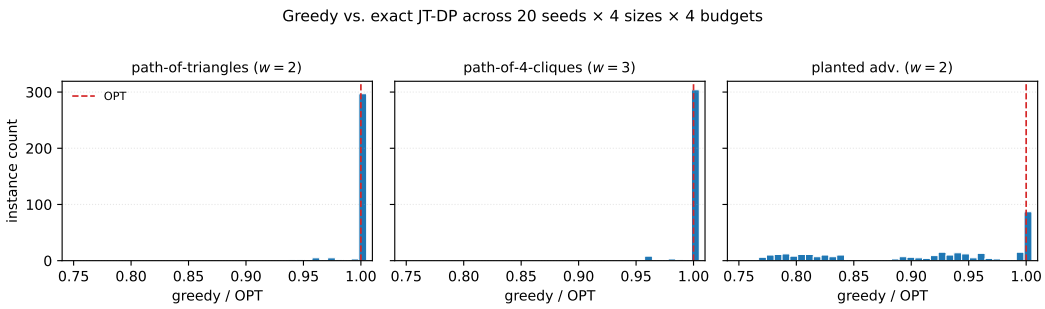


Figure 4: Distribution of greedy/OPT ratios across 960 instances. Dashed red line marks OPT. The planted adversarial family (right panel) shows suboptimal greedy performance. The other families are predominantly greedy-optimal.

Table 2: Budget-splitting DP on tree-structured DSFs (d=2, mean $\pm$ std over 20 seeds). Generic $O(k^2)$ and accelerated $O(k)$ return identical values to machine precision. Speedup is the ratio of the mean generic and accelerated times.

n	k	d	generic (ms)	accelerated (ms)	speedup
100	50	2	11.5 ± 0.4	1.42 ± 0.02	8.1×
500	50	2	111.5 ± 63.3	16.51 ± 8.67	6.8×
200	100	2	83.5 ± 2.1	3.44 ± 0.29	24.3×
500	100	2	270.5 ± 47.4	11.15 ± 2.26	24.3×

Both methods return identical objective values within machine precision across all 720 instances (maximum gap 4.3×10^{-14}).

O.7 Extended convolution scaling

We compare brute-force $O(k^2)$ and marginal-merge $O(k)$ convolution on random discretely concave sequences for $k \in \{50, 100, 200, 500, 1000, 2000, 5000\}$ over 20 seeds. The speedup reaches 1251× at $k = 5000$. This acceleration requires concave budget tables, as established for the computation-tree regime in Lemma K.2. For generic bounded-width JT-DP, message concavity is an additional condition (Section K.4).

For $k \geq 200$, log-log fits to the mean runtimes have slopes 2.19 (brute force) and 1.20 (marginal merge). Figure 6 annotates fits to the minimum across seeds at each budget, with slopes 2.11 and 1.07, respectively.

Instance generation. Each concave budget table is generated by sampling k nonnegative marginal increments from $\mathrm{Uniform}(0, 1)$, sorting them in decreasing order, and taking the prefix sum. This produces a non-decreasing, discretely concave sequence by construction.

Numerical precision. The maximum discrepancy between brute-force and marginal-merge outputs across all 280 method runs is 3.0×10^{-11}, well below the 10^{-10} correctness threshold.

O.8 Feature selection with learned tractable objectives

Shared protocol. HAR and ISOLET use their official subject-disjoint and speaker-disjoint splits, with 561 and 617 features, respectively (Reyes-Ortiz et al., 2012; Cole & Fanty, 1994). A supplied feature is one ground-set item. Feature relevance $r_i \geq 0$ is estimated by three-nearest-neighbor mutual information with the label. We select each model's parameters separately for each budget by mean validation accuracy over three subject-grouped folds for HAR and four speaker-grouped folds for ISOLET. Constant-feature filtering, relevance, correlations, and support structures are fitted

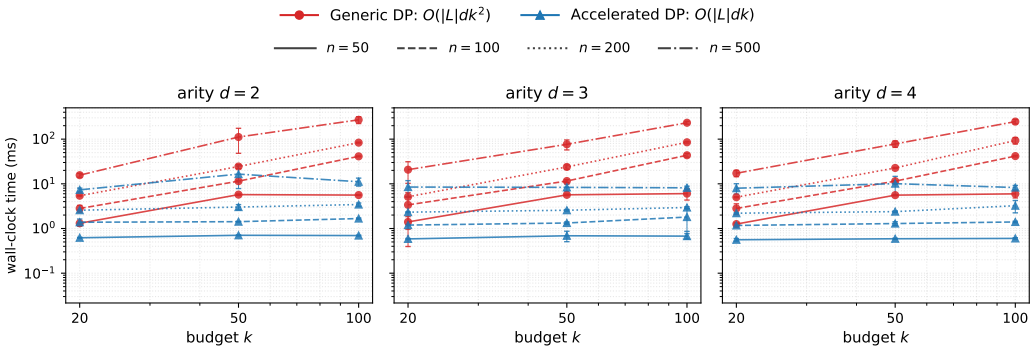


Figure 5: Budget-splitting DP acceleration across arities $d \in \{2, 3, 4\}$ (mean $\pm$ std, 20 seeds).

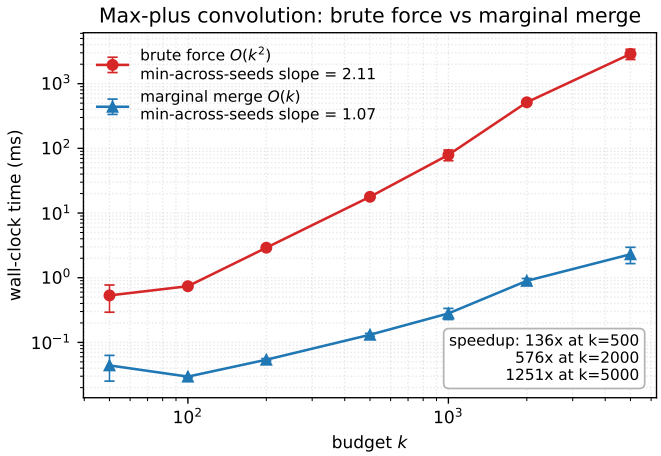


Figure 6: Max-plus convolution runtime (mean $\pm$ std over 20 seeds). Curves show mean times. Annotated slopes fit the minimum across seeds at each $k \geq 200$. Marginal merge agrees with brute force to numerical precision.

within each training fold. All methods use a standardized linear SVM with $C = 1$, with scaling fitted on the corresponding training data. The estimators use seed 20260726. Selected configurations are refitted on the full training split, and the resulting feature sets and classifiers are fixed before test evaluation. The official splits were also used in earlier experiments in this study. Table 3 gives all six budgets in the learned-cluster comparison.

Table 3: Feature-selection test accuracy (%). Random is the mean over 20 tuned partitions.

Dataset	k	Modular	Tree greedy	Tree exact	Group caps	Dense pairwise	Random
HAR	20	78.59	82.29	82.83	88.53	87.55	81.55
HAR	40	85.07	89.99	89.68	85.07	85.07	86.50
HAR	80	90.53	91.55	91.55	92.64	90.53	91.25
ISOLET	25	63.50	69.53	68.31	81.91	82.42	63.82
ISOLET	43	70.37	78.32	77.36	88.07	89.03	73.09
ISOLET	86	79.54	88.26	89.03	90.83	92.11	87.64

Tree and dense pairwise objectives. Pair dependence is $q_{ij} = -\frac{1}{2}\log(1-\rho_{ij}^2)$, with $|\rho_{ij}|$ clipped below one. For an edge set E, define

$$Q_E(A) = \sum_{i\in A} r_i - \sum_{\{i,j\}\in E} p_{ij}\mathbf{1}[i,j\in A], \qquad p_{ij} = \frac{2\beta q_{ij}}{k-1}.$$

The tree uses a maximum-weight spanning tree on q_{ij} and is optimized by exact DP. The dense baseline uses all pairs and is optimized by greedy. Each independently selects $\beta \in \{0, 0.25, 0.5, 1, 2, 4, 8\}$, breaking validation ties toward smaller β. Tree greedy uses the same fitted tree and selected β as tree exact.

For either edge set, choose $\gamma \geq \max\{0, \max_i(\sum_{j:\{i,j\}\in E} p_{ij} - r_i)\}$ and put $b_i = r_i + \gamma - \sum_{j:\{i,j\}\in E} p_{ij} \geq 0$. Then

$$Q_E(A) + \gamma|A| = \sum_i b_i\mathbf{1}[i\in A] + \sum_{\{i,j\}\in E} p_{ij}\min(|A\cap\{i,j\}|, 1).$$

This is a monotone depth-1 DSF with the same fixed-cardinality maximizers as Q_E. Its support graph is a subgraph of E, giving width at most one for the tree model. The tree decomposition passes factor-containment and running-intersection checks. Specialized tree DP agrees with generic JT-DP and exhaustive search on small induced instances.

For the ISOLET topology controls, uniform labeled trees are sampled by uniform Prüfer sequences. Degree-matched trees use shuffled Prüfer multisets, with edge weights rescaled to the learned tree's total dependence weight. Each control independently selects β by the same validation procedure. Averaged over the three reported budgets, the learned topology outperforms all 20 controls in each family.

Learned disjoint groups. We cluster features using distance $1-|\rho_{ij}|$. The candidate grid combines average or complete linkage, cut heights $0.1, 0.2, 0.4$, and $\alpha \in \{1, 2, 4\}$ in F_{cap}, together with a modular candidate, for 19 configurations. Exact validation ties prefer the modular candidate, then larger α, average linkage, and smaller cut height. Disjoint groups give a computation tree with capped leaf ridges and an identity output root, satisfying Proposition J.1. Budget merging determines the selected sets used for downstream evaluation. Validation selects average linkage at cut 0.4 with $\alpha = 1$ (119 groups) at all three ISOLET budgets; on HAR it selects average linkage at cut 0.1 with $\alpha = 2$ (224 groups) at $k = 20$, the modular candidate at $k = 40$, and complete linkage at cut 0.1 with $\alpha = 2$ (247 groups) at $k = 80$.

For each candidate partition, 20 random controls permute feature memberships while preserving group sizes. Caps are recomputed from the new memberships. Every replica independently selects among the same 19 configurations by grouped validation. The random mean in Table 3 averages their individual test accuracies.

Paired 95% percentile intervals use 10,000 subject or speaker bootstrap resamples, with seed 20260924. For each resample, the difference of correct prediction counts is divided by its total number of examples. The random comparator averages per-example correctness over the 20 fitted models before resampling. Intervals condition on the fitted models and are unadjusted for multiple comparisons. Table 4 reports comparisons with random grouping and the dense pairwise objective.

Table 4: Group caps minus comparator accuracy (pp, paired 95% intervals).

Dataset	k	Random mean	Dense pairwise
HAR	20	$+6.98$ $[+4.61, +10.15]$	$+0.98$ $[-1.11, +3.43]$
HAR	40	-1.43 $[-2.94, -0.38]$	$+0.00$ $[+0.00, +0.00]$
HAR	80	$+1.39$ $[+0.06, +3.35]$	$+2.10$ $[+0.46, +4.51]$
ISOLET	25	$+18.09$ $[+15.62, +20.49]$	-0.51 $[-2.37, +1.35]$
ISOLET	43	$+14.98$ $[+12.89, +17.01]$	-0.96 $[-2.57, +0.71]$
ISOLET	86	$+3.19$ $[+1.54, +4.69]$	-1.28 $[-2.57, +0.00]$

Selection cost. Greedy, lazy greedy, budget merging, and a heap that merges sorted group marginals agree in objective value at every tested prefix budget, as predicted by Corollary 4.4. On HAR and

ISOLET, median selection times are below 5 ms for all four solvers, with direct marginal-gain greedy faster than budget merging. These measurements include sorting and selected-set recovery and exclude objective fitting.

O.9 CIFAR-100 TAXONOMY-AWARE DATA SELECTION

We build the HC-caps objective as a three-level DSF over the CIFAR-100 hierarchy (root $\to$ 20 superclasses $\to$ 100 classes $\to$ 50 000 images) and select subsets under budgets $k \in \{500, 1000\}$.

HC-caps objective. Leaf weights are nonnegative-clipped cosine similarities between each image's frozen feature vector and its class centroid. Features are the 512-dimensional vectors after global average pooling in an ImageNet-pretrained ResNet-18, with the classifier head removed and all weights frozen. Class centroids are computed from the CIFAR-100 training split only. Activations at each internal node are $\varphi_n(t) = \min(t, c_n)$ with budget-scaled caps: $c_{\text{class}} = \lfloor k/100 \rfloor$, $c_{\text{super}} = \lfloor k/20 \rfloor$, and $c_{\text{root}} = k$.

The comparisons below use the same frozen features and downstream training protocol for HC-caps and its controls. Exact DP and priority-queue lazy greedy optimize the same HC-caps objective. The objective is designed from the supplied taxonomy and is fixed throughout each comparison.

Training protocol. ResNet-18 is trained from scratch on each selected subset for 200 epochs using SGD with momentum 0.9, weight decay 5×10^{-4}, initial learning rate 0.1 with cosine decay, batch size 128, random crop, horizontal flip, and standard CIFAR normalization. The paired comparisons and their sample sizes are reported below.

O.10 CIFAR-100 FACILITY-LOCATION COMPARISONS

Table 5 reports the paired accuracy differences at $k = 500, 1000$.

Let $s_{ij} = \max\{0, \cos(z_i, z_j)\}$ for frozen embeddings z_i. Global FL maximizes $\sum_i \max_{j \in A} s_{ij}$. Class-balanced FL applies the same criterion within each fine class and selects $k/100$ items per class. Taxonomy-matched FL averages three normalized coverage scores: global, mean superclass coverage, and mean fine-class coverage. Within each block, both targets and selected representatives are restricted to that block and coverage is divided by its size. All FL controls use priority-queue lazy greedy.

Accuracy differences pair HC-caps and each comparator by training seed within budget. Each comparison uses five paired runs, except taxonomy-matched FL at $k = 1000$, which uses ten. Each interval is $\bar{\delta} \pm t_{0.975,r-1} s_\delta / \sqrt{r}$, computed from that comparator's r paired differences. Selection times are single CPU wall-clock measurements of the solver calls. Global and taxonomy-matched FL include initial-bound computation but exclude construction of the dense similarity matrix. Class-balanced FL excludes construction of its class similarity blocks. HC-caps timing begins after loading the objective.

Table 5: HC-caps minus comparator test accuracy (pp, paired 95% CI). Five pairs were used except for taxonomy-matched FL at $k = 1000$ (ten).

Comparator	$k = 500$	$k = 1000$
Global FL	-0.60 $[-1.41, 0.22]$	0.28 $[-0.25, 0.81]$
Class-balanced FL	-0.42 $[-0.99, 0.15]$	-0.05 $[-0.38, 0.28]$
Taxonomy-matched FL	0.03 $[-0.94, 1.00]$	0.18 $[-0.13, 0.49]$
Uniform random	2.81 $[2.02, 3.60]$	3.06 $[2.77, 3.35]$
Balanced random	2.98 $[2.20, 3.76]$	2.87 $[2.39, 3.34]$

All six FL intervals at these budgets include zero. DP and same-objective priority-queue lazy greedy match in objective value, while DP is $24.4\times$ and $16.8\times$ faster at $k = 500$ and 1000, respectively. Relative to the FL controls, the reported selection-time ratios are approximately 13–$484\times$. These timings quantify the selection cost for objectives evaluated under the same downstream training protocol.

Table 6: Selection time (s). Global and taxonomy-matched FL times exclude one-off dense-matrix construction.

Method	$k = 500$	$k = 1000$
HC-caps exact DP	0.040	0.056
HC-caps priority-queue lazy greedy	0.975	0.940
Global FL lazy greedy	14.835	17.281
Class-balanced FL lazy greedy	0.713	0.749
Taxonomy-matched FL lazy greedy	19.367	21.895

Effective form of the objective. For the reported HC-caps budgets, five fine classes per superclass imply $5c_{\text{class}} \leq c_{\text{super}}$, and $20c_{\text{super}} \leq c_{\text{root}}$. With unit inter-node weights, summed child outputs never exceed their parent caps. Hence the superclass and root activations do not bind, and the set function reduces to a sum of fine-class capped factors. The resulting disjoint class supports place this objective in the exact budget-splitting regime. Removing the fine-class caps yields modular top-k, covering only 46/63 classes at $k = 500/1000$, versus 100/99 for HC-caps. The class caps therefore materially change the coverage of the selected subset.