CONFORMAL PREDICTION SETS FOR PERMUTATIONS AS BIPARTITE MATCHING GRAPHS

Anonymous authors
Paper under double-blind review

ABSTRACT

Permutation prediction sets may contain factorially many complete assignments, making explicit enumeration unusable even when statistical calibration is straightforward. We study how to represent such sets exactly while preserving the one-to-one constraint. A bottleneck score assigns a permutation the nonconformity of its worst selected edge; at every threshold, the accepted permutations then coincide exactly with the perfect matchings of a bipartite graph of admissible edges. The graph uses at most n^2 edges and supports membership, feasibility, representative-assignment, feasible-partner, and localization queries without enumerating its perfect matchings. Split conformal calibration selects the graph threshold and gives finite-sample marginal coverage of the complete assignment under exchangeability. We characterize the scope of this representation: every threshold set of a permutation score is exactly graph-representable if and only if the score has a bottleneck form. Across synthetic, sorting, and frozen pretrained keypoint-matching experiments, the method achieves near-nominal coverage and exposes localized and global ambiguity. In sorting, roughly 149 edges represent a median 2.99×10^{12} accepted permutations.

1 INTRODUCTION

Prediction sets over combinatorial output spaces pose a representational problem: a problem of modest input size can admit factorially many feasible outputs. One-to-one matching is a canonical example. Given two sets of objects, a permutation encodes a complete correspondence rather than a collection of independent labels, and assigning one row object restricts the assignments available to every other row. Applications include matching landmarks across images, aligning cells across modalities, matching detected objects across frames, entity resolution across databases, and recovering a canonical ordering from noisy observations.

Matching models commonly return only the highest-scoring permutation from an assignment score matrix (Kuhn, 1955; Mena et al., 2018; Zanfir & Sminchisescu, 2018), hiding plausible alternatives when objects are similar. The useful uncertainty question is often local: which partners for an item remain compatible with a globally valid assignment? Explicit lists require generating and searching candidate permutations, while independent per-item alternatives can violate the one-to-one constraint. We therefore seek an implicit representation that supports local queries, preserves assignment dependencies, and certifies coverage of the complete correspondence.

Conformal prediction uses held-out examples to calibrate a set with distribution-free marginal coverage under exchangeability (Vovk et al., 2005; Angelopoulos & Bates, 2023). It can be applied to any scalar score on a complete permutation. The challenge addressed here is different: how to represent and query the resulting structured set without treating all $n!$ permutations as separately enumerated output classes. Conformal calibration answers which member of a nested set family to return; the representation determines how that entire set is stored and queried. Recent work addresses related difficulties for structured and generative outputs using sampled candidates, ranked structured solutions, or implicit set families (Quach et al., 2024; Cauchois et al., 2024; Zhang et al., 2025; Gao et al., 2025; Gollapudi et al., 2026).

Our key observation is that matching feasibility itself can represent the set. Thresholding row–column edges produces a bipartite graph, and a bottleneck score makes its perfect matchings coincide exactly with the accepted permutations: none are removed or added. Retained edges express

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.

local admissibility, but every accepted output remains a globally valid one-to-one assignment. Split conformal calibration selects the threshold for finite-sample marginal coverage of the complete assignment.

The graph supports candidate membership, feasibility, representative prediction, and feasible-partner queries without enumerating the set; each reported partner extends to a perfect matching. Its connected components distinguish globally coupled ambiguity from uncertainty confined to a few object groups.

Representation size is distinct from set cardinality. A graph stores at most n^2 edges even when it represents factorially many matchings, but a short enumerated list can be smaller on easy, small instances. We therefore distinguish the number of accepted permutations, storage and query cost, and localization throughout the empirical analysis.

Our contributions are: (i) an exact implicit representation of permutation-valued prediction sets as the perfect matchings of a bipartite graph, together with direct algorithms for membership, feasibility, representative prediction, feasible partners, and localization; (ii) a characterization showing that every threshold set of a permutation score is exactly graph-representable if and only if the score has a bottleneck form, together with the resulting expressivity tradeoffs; and (iii) a split-conformal instantiation with finite-sample marginal coverage of the complete assignment, component factorization, and experiments demonstrating scalability and localized or global ambiguity.

More broadly, the construction illustrates a design principle for combinatorial uncertainty: represent complete outputs through admissible local primitives together with the original global feasibility constraints. Our formal results establish this principle only for permutations and bipartite perfect matchings; extending exact representation and queryability guarantees to other structured families is a direction for future work.

2 Related Work

Conformal prediction and structured outputs. Existing approaches to structured conformal prediction differ in how they search for, approximate, or represent the accepted output set. Conformal prediction converts a fixed score into a prediction set with finite-sample marginal coverage under exchangeability (Vovk et al., 2005; Lei et al., 2018; Angelopoulos & Bates, 2023). For combinatorial outputs, prior work calibrates sampling and rejection for free-form generations (Quach et al., 2024), uses sequential M-best search for matching (Cauchois et al., 2024), optimizes over interpretable set families (Zhang et al., 2025; Gao et al., 2025), or approximately compresses route collections into subgraphs (Gollapudi et al., 2026). Our distinction is representational rather than a different coverage principle: for bottleneck permutation scores, one graph represents the entire threshold set exactly without first enumerating candidate permutations.

Permutation learning and assignment uncertainty. Differentiable permutation and graph-matching systems produce affinity matrices and enforce correspondence with Sinkhorn normalization or matching solvers (Mena et al., 2018; Zanfir & Sminchisescu, 2018; Wang et al., 2024); we treat the matcher as fixed. Ranked assignment, diverse-M-best, and perturb-and-MAP methods summarize uncertainty with high-scoring solutions (Murty, 1968; Ramakrishna & Batra, 2012), while exact normalization and marginals for edge-factorized perfect matchings involve a #P-hard permanent (Valiant, 1979). Classical bottleneck assignment minimizes the largest selected-edge cost (Derigs & Zimmermann, 1978). We use this objective to define nested sets of complete assignments: split conformal selects the threshold, and one graph represents every accepted assignment exactly. Matching and allowed-edge algorithms supply the queries (Hopcroft & Karp, 1973; Dulmage & Mendelsohn, 1958; Tassa, 2012); ordinary connected components factor the represented uncertainty.

3 Problem Setup

For an input x, let $A = \{a_1, \ldots, a_n\}$ and $B = \{b_1, \ldots, b_n\}$ be two sets to match. Let S_n denote the set of all $n!$ permutations of n items. A permutation $\pi \in S_n$ represents the complete one-to-one assignment $a_i \mapsto b_{\pi(i)}$; the ground-truth assignment is $\pi^\star$.

We assume a fixed matcher returns model scores $W(x) \in \mathbb{R}^{n\times n}$, where larger $W_{ij}(x)$ favors the edge $i \mapsto j$. This includes log-probabilities $W = \log P$ and unnormalized model scores; no stochasticity is required.

The usual single best assignment (point prediction) is

$$\widehat{\pi}(x) = \arg\max_{\pi\in S_n} \sum_{i=1}^{n} W_{i,\pi(i)}(x).$$

Our objective is to construct a set-valued predictor $\mathcal{C}(x) \subseteq S_n$, such that, for a user-specified error level $\alpha \in (0, 1)$,

$$\Pr\{\pi^\star \in \mathcal{C}(X)\} \geq 1 - \alpha.$$

The event covers the entire ground-truth permutation, not its rows separately; the probability is over the calibration data and a fresh test example.

4 Conformal Sets as Matching Graphs

We first construct a nested family of permutation sets represented exactly by bipartite graphs. A bottleneck score gives an equivalent scalar-score view of the same family, allowing held-out conformal calibration to select its operating threshold. The calibrated graph then supports matching-specific queries and reveals whether uncertainty is local or globally coupled.

4.1 Exact Graph Representation

We seek an implicit object whose feasible solutions are exactly the accepted complete permutations. Define the edge nonconformity score

$$s_{ij}(x) = -W_{ij}(x),$$

so smaller values indicate edges the model regards as more plausible. If $W = \log P$, then $s_{ij} = -\log P_{ij}$; otherwise s_{ij} is simply the negative unnormalized model score. For any threshold t, retain the admissible edges

$$E_t(x) = \{(r_i, \ell_j) : s_{ij}(x) \leq t\}$$

in the bipartite graph $\mathcal{G}_t(x) = (R \cup L, E_t(x))$, where $R = \{r_1, \ldots, r_n\}$ and $L = \{\ell_1, \ldots, \ell_n\}$ are the row and column vertices. Here "admissible" means only that the edge score is at most t. Let $\mathcal{M}(E_t(x))$ denote all perfect matchings using retained edges, equivalently all complete legal one-to-one assignments supported by the graph. We define

$$\mathcal{C}_t(x) = \mathcal{M}(E_t(x)) = \{\pi \in S_n : (r_i, \ell_{\pi(i)}) \in E_t(x)\ \forall i\}.$$

Thus a graph with at most n^2 edges represents the accepted set exactly.

Split conformal calibration requires one scalar score for each labeled complete output. A permutation π selects the n scores $s_{i,\pi(i)}(x)$, so its graph membership satisfies

$$\pi \in \mathcal{M}(E_t(x)) \quad\Longleftrightarrow\quad s_{i,\pi(i)}(x) \leq t\ \ \forall i \quad\Longleftrightarrow\quad \max_i s_{i,\pi(i)}(x) \leq t.$$

This motivates the complete-permutation bottleneck score

$$S(x, \pi) = \max_i s_{i,\pi(i)}(x) = -\min_i W_{i,\pi(i)}(x),$$

the nonconformity of the worst edge used by π. The two views of the prediction set coincide:

$$\underbrace{\{\pi \in S_n : S(x, \pi) \leq t\}}_{\text{score-defined threshold set}} = \underbrace{\mathcal{M}(E_t(x))}_{\text{perfect matchings of the graph}}$$

Theorem 1 later establishes the converse: exact graph representation of every threshold set implies a bottleneck form.

4.2 CONFORMAL CALIBRATION

We use held-out calibration data $\mathcal{D}_{\text{cal}} = \{(x_k, \pi_k)\}_{k=1}^{m}$, not used to train the matcher. As in split conformal prediction, these labeled examples and the future test example are assumed exchangeable; i.i.d. sampling suffices. For calibration scores $S_k = S(x_k, \pi_k)$, let $S_{(1)} \le \cdots \le S_{(m)}$ and set

$$r_\alpha = \lceil (m+1)(1-\alpha) \rceil, \qquad q_\alpha = \begin{cases} S_{(r_\alpha)}, & r_\alpha \le m, \\ +\infty, & r_\alpha > m. \end{cases}$$

Here S_k is the worst true-edge nonconformity in example k: a threshold at least S_k is necessary and sufficient to retain that example's complete ground-truth permutation. The quantile above is the standard split-conformal choice (Vovk et al., 2005; Angelopoulos & Bates, 2023); if $r_\alpha = m+1$, the calibration sample is too small to certify the requested coverage with a nontrivial finite threshold, so $q_\alpha = +\infty$. The calibrated output is $\mathcal{G}_\alpha(x) = \mathcal{G}_{q_\alpha}(x)$, and its represented set is

$$\mathcal{C}_\alpha(x) = \mathcal{C}_{q_\alpha}(x) = \{\pi \in S_n : S(x, \pi) \le q_\alpha\}.$$

The graph retains $s_{ij}(x) \le q_\alpha$, equivalently $W_{ij}(x) \ge -q_\alpha$; in the probability special case this is $P_{ij}(x) \ge \exp(-q_\alpha)$. Only ground-truth calibration permutations are scored, so calibration requires neither candidate generation nor permutation enumeration. Calibration learns one scalar cutoff, while each test input produces a different graph through its score matrix. Proposition 1 gives marginal coverage; a pooled cutoff does not imply equal coverage conditional on problem size or other covariates. Algorithm 1 in Appendix D summarizes the procedure as selection of one member of the nested, exactly represented family.

4.3 QUERIES WITHOUT ENUMERATION

Once the prediction set is represented as a graph, common questions can be answered without listing its perfect matchings. Candidate membership takes $O(n)$ edge lookups, and nonemptiness is a bipartite perfect-matching test. When $\mathcal{G}_t(x)$ is feasible, a representative is obtained by solving a maximum-weight perfect matching restricted to $E_t(x)$, using the original model scores as weights. If it is infeasible, then $\mathcal{C}_t(x) = \emptyset$, no representative exists, and the feasible-partner sets below are empty.

The feasible partners of row i are

$$\mathcal{F}_{i,t}(x) = \{j : (r_i, \ell_j) \in E_t(x) \text{ and } \mathcal{G}_t(x) \setminus \{r_i, \ell_j\} \text{ has a perfect matching}\}.$$

In words, j is feasible for row i if fixing $i \mapsto j$ leaves a perfect matching for the remaining rows and columns. A retained edge need not be globally feasible. In a feasible graph, the feasible-partner edges are precisely its *allowed edges*, those belonging to at least one perfect matching. Given one perfect matching M, an edge is allowed if it belongs to M or lies on an M-alternating cycle; strongly connected components of an auxiliary directed graph identify all such edges jointly (Tassa, 2012). With adjacency lists, this takes $O(n+|E_t|)$ additional time after finding M, which costs $O(|E_t|\sqrt{n})$ using Hopcroft–Karp (Hopcroft & Karp, 1973). Our SPair evaluation uses the same criterion with dense arrays and a dense assignment solver, computing the support once per graph.

4.4 LOCAL STRUCTURE OF UNCERTAINTY

Is uncertainty spread across the entire assignment, or confined to a few local groups? Decompose $\mathcal{G}_t(x) = \mathcal{G}_1(x) \cup \cdots \cup \mathcal{G}_M(x)$ into connected components, where $\mathcal{G}_\ell$ has row vertices R_ℓ, column vertices L_ℓ, and edges E_ℓ. No admissible edge joins distinct components.

Let $\mathcal{M}_\ell(x)$ be the perfect matchings of $\mathcal{G}_\ell$. A component is *forced* when $|\mathcal{M}_\ell(x)| = 1$ and *ambiguous* otherwise. A forced component need not be a single edge: it may contain extra edges that cannot participate in an alternative perfect matching. Because no edge crosses components, a global accepted permutation is formed by choosing one perfect matching in each component and taking their union. When the graph is feasible,

$$\mathcal{C}_t(x) \cong \mathcal{M}_1(x) \times \cdots \times \mathcal{M}_M(x).$$

Thus deterministic regions can be reported as fixed assignments, while ambiguous alternatives can be inspected in their local subgraphs. Proposition 4 formalizes this set factorization.

5 GUARANTEES AND REPRESENTATION

5.1 COVERAGE GUARANTEE

The coverage event below concerns the entire ground-truth permutation, not individual rows or edges.

Proposition 1 (Split conformal coverage)**.** *Assume the m calibration examples and the test example are exchangeable. Let q_α be the split-conformal quantile defined above. Then*

$$\Pr\{\Pi_{m+1} \in \mathcal{C}_\alpha(X_{m+1})\} \geq 1 - \alpha.$$

If in addition the scores $S(X_k, \Pi_k)$, $k = 1, \ldots, m+1$, are almost surely distinct, then

$$\Pr\{\Pi_{m+1} \in \mathcal{C}_\alpha(X_{m+1})\} = \frac{r_\alpha}{m+1} \;\in\; \left[1 - \alpha,\; 1 - \alpha + \tfrac{1}{m+1}\right).$$

The standard exchangeability rank argument proves the result. Because an empty set necessarily misses the truth, Proposition 1 also gives $\Pr\{\mathcal{C}_\alpha(X_{m+1}) = \emptyset\} \leq \alpha$; Appendix H gives the formal proof. Under no ties, marginal coverage is exactly $r_\alpha/(m+1)$; Appendix G gives the stronger i.i.d.-continuous calculation used to contextualize variation across experimental splits, while ties explain why the discrete `worst_rank` ablation over-covers.

5.2 EXACT GRAPH REPRESENTATION

The coverage guarantee above holds for any nonconformity score. What distinguishes the bottleneck score is representational: its accepted set can be recovered exactly from local edge admissibility and the original one-to-one constraint. For a score S and threshold t, we call $\mathcal{C}_t = \{\pi : S(\pi) \leq t\}$ its *threshold set*.

Fix x and abbreviate $S(\pi) = S(x, \pi)$. For an edge set E, let $\mathcal{M}(E) = \{\pi \in S_n : (i, \pi(i)) \in E \; \forall i\}$ be its perfect matchings. A graph records which edges may be used, but not which accepted permutations those edges came from. Collecting every used edge may therefore create new permutations by recombining edges from different members of the original set. Formally, define the *support*

$$\mathrm{supp}(\mathcal{C}) = \{(i, j) : \pi(i) = j \text{ for some } \pi \in \mathcal{C}\}.$$

Definition 1 (Graph-representable set)**.** *A set of permutations $\mathcal{C} \subseteq S_n$ is* graph-representable *if there exists an edge set $E \subseteq \{1, \ldots, n\}^2$ with $\mathcal{C} = \mathcal{M}(E)$. The representation is exact: it neither removes a member of $\mathcal{C}$ nor adds a new permutation.*

The following result says that exact representation is possible precisely when forgetting edge co-occurrence and allowing all legal recombinations adds nothing.

Proposition 2 (Set-level characterization)**.** *For any $\mathcal{C} \subseteq S_n$ we have $\mathcal{C} \subseteq \mathcal{M}(\mathrm{supp}(\mathcal{C}))$, and $\mathcal{C}$ is graph-representable if and only if*

$$\mathcal{C} = \mathcal{M}(\mathrm{supp}(\mathcal{C})).$$

In that case $E = \mathrm{supp}(\mathcal{C})$ is the smallest edge set representing $\mathcal{C}$.

For an arbitrary set, $\mathcal{M}(\mathrm{supp}(\mathcal{C}))$ is the smallest graph-representable superset obtained after forgetting which edges originally co-occurred. Exact representability holds precisely when this operation adds no permutation. The proof is in Appendix F.

5.3 WHEN CAN THRESHOLD SETS BE REPRESENTED?

Imagine increasing t. Each edge enters an admissible graph at an activation threshold, and a permutation first becomes feasible only after every edge it uses has entered. Its activation threshold is therefore the largest activation threshold among its selected edges. This suggests a maximum-over-edges form.

Call $S : S_n \to \mathbb{R} \cup \{+\infty\}$ a *bottleneck score* if there exists an edge function $s : \{1, \ldots, n\}^2 \to \mathbb{R} \cup \{+\infty\}$ such that $S(\pi) = \max_i s(i, \pi(i))$. The assignment is thus scored by its worst selected edge. Because q_α is data-dependent and α is user-chosen, we require representability at every threshold rather than at one operating point.

Theorem 1 (Characterization of graph-representable scores)**.** *Let $S : S_n \to \mathbb{R} \cup \{+\infty\}$ be any score. The following are equivalent.*

1. *Every threshold set $\mathcal{C}_t = \{\pi \in S_n : S(\pi) \le t\}$ is graph-representable.*
2. *S is a bottleneck score.*

Moreover, when these hold an edge function is recovered canonically from S as

$$s^\star(i,j) = \min_{\pi \in S_n:\ \pi(i)=j} S(\pi), \qquad \textit{and} \qquad S(\pi) = \max_i s^\star(i, \pi(i)).$$

Here $s^\star(i,j)$ is the best complete-permutation score achievable while forcing $i \mapsto j$, equivalently the earliest threshold at which that edge appears in an accepted permutation. It is a canonical edge function derived from S, not necessarily the model's original edge score.

The theorem separates statistical validity from representational structure. Many permutation scores can be conformalized, but every threshold set is represented exactly by edge admissibility precisely for bottleneck scores. Maximum aggregation is therefore implied by the desired representation, while the edge statistic remains a modeling choice; for example, the within-row-rank ablation `worst_rank` is also graph-representable. Appendix F gives the proof.

5.4 Expressivity Tradeoffs

Exact edge-based representation provides compact storage and matching-specific queries, but restricts which permutation subsets can be expressed. One tradeoff is size sensitivity.

Remark 1 (Size sensitivity)**.** *Let $Z_i = s_{i,\pi^\star(i)}(X)$ be the i-th true-edge score. For illustration, suppose that conditional on n, the Z_i are i.i.d. with a common CDF F that does not vary with n. Then*

$$\Pr\{S(X,\pi^\star) \le t \mid n\} = \Pr\{\max_{i \le n} Z_i \le t \mid n\} = F(t)^n.$$

For $0 < F(t) < 1$, this decreases with n: longer permutations provide more opportunities for one true edge to determine the maximum. A pooled split-conformal threshold still guarantees marginal coverage over the size mixture, but need not equalize coverage conditional on n. Applying split conformal separately within prespecified size groups gives group-conditional guarantees under within-group exchangeability, at the cost of smaller calibration groups; Appendix A.1 evaluates this tradeoff on SPair-71k.

The broader expressivity limit is that an edge-only graph cannot encode arbitrary higher-order restrictions. For example, two edges may each occur in some low-score permutation but be forbidden together. Once both edges are retained, an ordinary graph cannot remember that conjunction. Appendix F gives an explicit $n = 3$ witness. More generally, any proper graph-representable subset of S_n excludes at least $(n-1)!$ permutations, so these sets do not provide arbitrary fine cardinality control. Additive and regret scores illustrate this boundary.

Proposition 3 (Additive and regret scores)**.** *Let $n \ge 3$ and let s be an edge function. Define the additive score $S_{\mathrm{sum}}(\pi) = \sum_i s(i, \pi(i))$ and the regret score $S_{\mathrm{reg}}(\pi) = S_{\mathrm{sum}}(\pi) - \min_{\sigma \in S_n} S_{\mathrm{sum}}(\sigma)$. For Lebesgue-almost every $s \in \mathbb{R}^{n \times n}$, neither score has graph-representable threshold sets at every threshold: each admits a threshold whose accepted set is not the perfect-matching set of any bipartite graph.*

Any score-defined set $\mathcal{C}$ has a conservative graph superset $\mathcal{M}(\mathrm{supp}(\mathcal{C}))$. This superset preserves coverage but can add score-rejected permutations; Theorem 1 identifies when the relaxation is exact at every threshold. This is a representational statement, not a free computational reduction: forming the support of a general structured set may require difficult constrained searches. Appendix F proves the proposition and the relaxation claims.

5.5 Factorization Across Components

If the calibrated graph has a global perfect matching, every connected component contains equally many row and column vertices. Since no edge crosses components, global accepted permutations are in bijection with independent choices of one perfect matching per component.

Method	Coverage	Cand. Med.	Cand. P90	Edges
`max_edge`	88.7 ± 2.6	2.7	12.0	15.0
`adaptive_mass`	90.2 ± 1.0	1.0	10.7	–
`top_k`	88.8 ± 3.6	3.7	3.7	–
`sum_nll`	88.3 ± 3.1	10.7	14.7	–
`worst_rank`	91.7 ± 5.2	118.7	150.7	29.3
`independent_rows`	88.7 ± 2.6	66.7	1296.6	15.0

Table 1: Synthetic comparison at $n = 8$, $\alpha = 0.10$, over three seeds with $m = 500$ calibration and 300 test examples per seed; coverage is mean $\pm$ standard deviation across seeds. The no-ties finite-sample target is $451/501 = 90.02\%$ (Vovk et al., 2005). Under i.i.d. continuous scores, Beta calibration variation plus finite-test noise gives a 1.26-point standard error for the mean (Appendix G); all continuous-score entries lie within 1.1 such standard errors of the target. The `worst_rank` ablation takes at most n distinct values, and such ties make the conformal quantile conservative, which accounts for its higher coverage. Independent-row candidates include non-permutations.

Proposition 4 (Ambiguous component factorization)**.** *Let $\mathcal{G}_1, \ldots, \mathcal{G}_M$ be the connected components of $\mathcal{G}_\alpha(x)$, with row and column node sets R_ℓ and L_ℓ. If $\mathcal{C}_\alpha(x) \neq \emptyset$, then*

$$|R_\ell| = |L_\ell| \quad \forall \ell, \qquad \mathcal{C}_\alpha(x) \cong \prod_{\ell=1}^{M} \mathcal{M}_\ell(x), \qquad |\mathcal{C}_\alpha(x)| = \prod_{\ell=1}^{M} |\mathcal{M}_\ell(x)|.$$

This is a factorization of the accepted set, not a claim of probabilistic or posterior independence.

Proof. No admissible edge joins distinct components, so a global perfect matching pairs R_ℓ only with L_ℓ, implying equal sizes. Restriction maps a global perfect matching to one matching in each $\mathcal{M}_\ell$; conversely, the union of one matching per component is global. This bijection gives the product and cardinality identities. □

6 Experiments

Within each benchmark, competing constructions use identical scores, splits, and target levels; across benchmarks the matcher changes while the conformal wrapper remains fixed. We separately report full-assignment coverage, *candidate cardinality* (atomic outputs represented), *representation size* (stored graph edges or permutations), and localization because these quantities need not improve together.

Baselines. All prediction-set baselines use the same matcher scores, calibration splits, and target levels. At small n, `sum_nll` thresholds total negative log-likelihood, `top_k` calibrates ground-truth rank, and `adaptive_mass` calibrates a cumulative-mass prefix after full enumeration. For the scalable explicit-list comparison, we use an additive-regret threshold and a rank-calibrated top-K set, materialized by exact sequential assignment generation (Murty, 1968; Cauchois et al., 2024); Appendix B.1 gives definitions and implementation details. The `worst_rank` and `independent_rows` variants ablate the graph score and one-to-one constraint, respectively; Gumbel-Sinkhorn and `pygmtools` supply base scores rather than prediction-set baselines (Mena et al., 2018; Wang et al., 2024).

6.1 Synthetic Gumbel-Sinkhorn Benchmark

Each instance samples a permutation and noisy score matrix; true edges receive a hardness-varying signal before Sinkhorn normalization. The main comparison uses $n = 8$, 500 calibration and 300 test examples per seed, and three seeds; additional runs use $n = 9$ and implicit graph summaries at $n \in \{16, 32, 64\}$.

Adaptive mass has the smallest median set, so `max_edge` has no universal cardinality advantage. It does represent fewer valid assignments than the additive and rank-threshold alternatives. Identi-

Graph operations at 50% density (ms / MiB)					
n	Edges	Construction	Feasibility	Representative	Support
64	2,065	0.005/58.2	0.089/58.4	0.150/58.4	0.400/59.1
256	32,877	0.011/58.5	1.18/58.9	1.82/58.9	3.48/61.7
1,024	524,545	0.340/75.8	18.42/76.3	26.84/92.2	54.43/105.9
2,048	2,098,116	2.12/127.8	76.89/130.1	105.30/194.3	231.37/245.3

Exact enumeration (s / MiB)			
n	$n!$	Enumeration	Completed
8	40,320	0.028/63.0	5/5
9	362,880	0.271/125.1	5/5
10	3,628,800	2.94/818.3	5/5
11	39,916,800	$--/3,769.0$	0/5

Table 2: Median time and peak memory for graph operations at 50% density and exact adaptive-mass enumeration. Support extracts all edges belonging to a perfect matching, including the initial matching solve. All $n = 11$ enumeration runs reached the 4 GiB limit.

Regime	n	Coverage	Set Med.	Edges	Components	Amb. P90	Max Local Med.
Near ties	20	90.1 ± 0.5	2,880	47.5	8.1	6	8
Blocks of 4	20	90.0 ± 0.6	7,962,624	78.9	5.0	4	24
Blocks of 5	20	89.7 ± 0.8	207,360,000	98.7	4.0	5	120
Blocks of 5	30	90.1 ± 1.0	2.99×10^{12}	149.1	5.8	10	120

Table 3: Controlled sorting at $\alpha = 0.10$. Coverage is mean percent $\pm$ standard deviation across eight seeds; the no-ties target is $1801/2001 = 90.005\%$. Amb. P90 is the 90th percentile largest ambiguous component, and Max Local Med. is its median matching count. Appendix G gives a 0.31-point reference standard error for mean coverage.

cal edges and coverage yield medians 2.7 versus 66.7 for `max_edge` and `independent_rows`, isolating the effect of enforcing distinct columns.

Runtime and memory scaling. We benchmark controlled matrices with independent uniform entries, a planted matching, and expected densities 10%, 50%, or 90%. Isolated single-threaded processes run on an Intel Xeon Gold 6338 CPU, with peak memory including Python and the input matrix. Graph entries use three process replicates, each summarizing ten generated matrices after warm-up; exact enumeration uses five cold processes with a 4 GiB limit and ten-minute timeout.

Exact enumeration requires 818 MiB and 2.94 seconds at $n = 10$ and fails at $n = 11$. At $n = 2048$, a 50%-dense graph stores 2.10 million edges over a label space with $\log_{10}(2048!) \approx 5894$: construction, feasibility, representative assignment, and all-partner support take 2.12, 76.89, 105.30, and 231.37 ms, respectively, with peak process memory of 245.3 MiB. Results remain practical across 10%–90% density (Appendix B.2). This table measures absolute scalability; Table 5 compares output-sensitive explicit lists. At $n = 9$, adaptive mass nevertheless returns a smaller median set (1.0 versus 8.2), so this is not a universal cardinality advantage.

6.2 Controlled Sorting and Component Localization

We sort noisy scalars under scattered near-ties or blocks of four or five nearly equal values, creating many orderings with known local ambiguity. Each setting uses eight seeds, 2,000 calibration examples, and 3,000 test examples per seed.

With five size-four blocks at $n = 20$, about 79 edges represent a median $24^5 = 7{,}962{,}624$ permutations as five components with only 24 local matchings each. At $n = 30$, the median set reaches 2.99×10^{12}, while the median largest local set has 120 matchings. The occasional 10-row component shows that localization is measured, not assumed.

6.3 Frozen Pretrained Matching on SPair-71k

We run released, frozen COMMON (Lin et al., 2023) on SPair-71k (Min et al., 2019). We capture its unnormalized scores $W = A$ immediately before Hungarian decoding and set $s_{ij} = -A_{ij}$, without Sinkhorn normalization or retraining. Intersection filtering and removal of mirrored directions give 6,117 square matching problems with $n \in [3, 20]$; five splits use 2,000 for calibration and 4,117 for test.

α	Target	Coverage	Exact	Density	Feasible	Partners
.05	95	95.24 ± 0.62	68.3	73.4	99.1	5.06
.10	90	90.53 ± 0.82	68.3	60.4	95.8	3.85
.20	80	80.24 ± 1.30	68.3	45.3	86.1	2.48

Table 4: Frozen COMMON on SPair-71k over five splits. Entries except Partners are percentages; Exact is point-permutation accuracy, Density is retained-edge density, and Feasible is nonempty-set frequency.

COMMON has 85.6% row accuracy but only 68.3% exact-permutation accuracy. At $\alpha = .10$, the graph gives 90.5% full-assignment coverage, 60.4% edge density, and 3.85 feasible partners per row. Pooled coverage varies with n; size-aware calibration improves the undercovered $n = 10$ and $n = 13$ groups while preserving overall coverage. Appendix A.1 gives the full tradeoff and query examples, and Appendix C gives controlled WillowObject and PascalVOC studies.

Data	Predictor	Coverage	Representation, median [P90]	Build	Support
SPair	Graph	90.53	25 [64] edges	0.004 ms	0.244 ms
	Additive-regret list	90.23	2 [18] assignments	0.521 ms	1.147 ms
	Rank-K list	90.53	10–11 assignments	1.750 ms	0.024 ms
Sorting, blocks of 4	Graph	90.04	80 [80] edges	0.003 ms	0.345 ms
	Additive-regret list	89.84	6.88M [7.96M] assignments	first 10k: 1.88 s	–
	Rank-K list	–	$\geq$ 5.61M assignments	–	–

Table 5: Output-sensitive explicit-list comparison at $\alpha = .10$. Graph size counts edges and list size counts complete assignments; predictors use the same scores, splits, and target level but define different sets. Sorting list sizes include certified lower bounds (Appendix B.1); none of 80 additive timing trials completed within 10,000 outputs.

On SPair, additive-regret lists contain only two assignments at the median and complete for every instance, so short lists are competitive. In block sorting, 99.0% of additive-regret sets exceed one million assignments and rank-K requires at least 5.58 million, while the graph uses about 79 edges. The graph's representational advantage is therefore strongest when local ambiguity produces many complete assignments; it is not universal compression.

7 Limitations and Discussion

The guarantee is marginal and relies on exchangeability and a fixed matcher. Calibration controls coverage; the matcher determines graph sparsity and localization. A sparse graph can still represent many permutations, and counting perfect matchings remains #P-hard (Valiant, 1979). Explicit lists can be smaller when few assignments are accepted; the graph avoids materialization when ambiguity produces many complete assignments. Bottleneck aggregation can be conservative, and pooling problem sizes does not ensure size-conditional coverage; size stratification can mitigate this but may require dense or complete graphs when calibration groups are small. The component product is a set factorization, not posterior independence. Appendix E discusses experimental scope and further qualifications.

8 Conclusion

A bottleneck score makes each permutation threshold set coincide exactly with the perfect matchings of a bipartite graph; split conformal calibration selects the threshold with finite-sample marginal coverage of the complete assignment. The graph preserves one-to-one validity and supports direct queries without enumerating its potentially factorially many members. Our characterization shows that maximum aggregation is necessary and sufficient for this exact representation at every threshold.

REPRODUCIBILITY STATEMENT

The appendix records data processing, calibration sizes, random seeds, output caps, and timing protocols. The accompanying code implements the algorithms and preserves aggregate and instance-level experiment outputs.

AI USE STATEMENT

We used generative AI tools to write, debug, and run the experiment code, including the method implementation, the synthetic-data simulator, and benchmark preprocessing, and to tabulate results; the authors designed all experiments. AI tools edited the wording and placement of author-written proofs but were not used to formulate claims, proof ideas, hypotheses, or conceptual frameworks, to design experiments, to translate, to analyze qualitative data, or to interpret results. They also helped search the literature, structure, draft, and edit the manuscript, and build the supplementary code. The authors reviewed and tested all AI-written code, verified every reported number with the reproduction script, checked all proofs by hand, and approved every text edit. We take full responsibility for this work, including AI-assisted text, claims, and artifacts.

REFERENCES

Anastasios N. Angelopoulos and Stephen Bates. A gentle introduction to conformal prediction and distribution-free uncertainty quantification. *Foundations and Trends in Machine Learning*, 16(4): 494–591, 2023. doi: 10.1561/2200000101.

Maxime Cauchois, Suyash Gupta, Alnur Ali, and John C. Duchi. Predictive inference with weak supervision. *Journal of Machine Learning Research*, 25(118):1–45, 2024. URL https://jmlr.org/papers/v25/23-0253.html.

Ulrich Derigs and Uwe Zimmermann. An augmenting path method for solving linear bottleneck assignment problems. *Computing*, 19(4):285–295, 1978. doi: 10.1007/BF02252026.

A. L. Dulmage and N. S. Mendelsohn. Coverings of bipartite graphs. *Canadian Journal of Mathematics*, 10:517–534, 1958. doi: 10.4153/CJM-1958-052-0.

Chao Gao, Liren Shan, Vaidehi Srinivas, and Aravindan Vijayaraghavan. Volume optimality in conformal prediction with structured prediction sets. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 18495–18527. PMLR, 2025. URL https://proceedings.mlr.press/v267/gao25k.html.

Sreenivas Gollapudi, Kostas Kollias, Kamesh Munagala, and Aravindan Vijayaraghavan. Compact conformal subgraphs. *arXiv preprint arXiv:2602.07530*, 2026. URL https://arxiv.org/abs/2602.07530.

John E. Hopcroft and Richard M. Karp. An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. *SIAM Journal on Computing*, 2(4):225–231, 1973. doi: 10.1137/0202019.

Harold W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1–2):83–97, 1955. doi: 10.1002/nav.3800020109.

Jing Lei, Max G'Sell, Alessandro Rinaldo, Ryan J. Tibshirani, and Larry Wasserman. Distribution-free predictive inference for regression. *Journal of the American Statistical Association*, 113 (523):1094–1111, 2018. doi: 10.1080/01621459.2017.1307116.

Yijie Lin, Mouxing Yang, Jun Yu, Peng Hu, Changqing Zhang, and Xi Peng. Graph matching with bi-level noisy correspondence. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023. URL https://arxiv.org/abs/2212.04085.

Gonzalo Mena, David Belanger, Scott Linderman, and Jasper Snoek. Learning latent permutations with Gumbel-Sinkhorn networks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=Byt3oJ-0W.

Juhong Min, Jongmin Lee, Jean Ponce, and Minsu Cho. SPair-71k: A large-scale benchmark for semantic correspondence. *arXiv preprint arXiv:1908.10543*, 2019. URL https://arxiv.org/abs/1908.10543.

Katta G. Murty. Letter to the editor—an algorithm for ranking all the assignments in order of increasing cost. *Operations Research*, 16(3):682–687, 1968. doi: 10.1287/opre.16.3.682.

Victor Quach, Adam Fisch, Tal Schuster, Adam Yala, Jae Ho Sohn, Tommi Jaakkola, and Regina Barzilay. Conformal language modeling. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=pzUhfQ74c5.

Varun Ramakrishna and Dhruv Batra. Mode-marginals: Expressing uncertainty via diverse M-best solutions. In *NeurIPS Workshop on Perturbations, Optimization, and Statistics*, 2012. URL https://mlp.cc.gatech.edu/papers/2012/rb_pos12/rb_pos12.pdf.

Tamir Tassa. Finding all maximally-matchable edges in a bipartite graph. *Theoretical Computer Science*, 423:50–58, 2012. doi: 10.1016/j.tcs.2011.12.071.

Leslie G. Valiant. The complexity of computing the permanent. *Theoretical Computer Science*, 8 (2):189–201, 1979. doi: 10.1016/0304-3975(79)90044-6.

Vladimir Vovk, Alexander Gammerman, and Glenn Shafer. *Algorithmic Learning in a Random World*. Springer, New York, 2005. doi: 10.1007/b106715.

Runzhong Wang, Ziao Guo, Wenzheng Pan, Jiale Ma, Yikai Zhang, Nan Yang, Qi Liu, Longxuan Wei, Hanxue Zhang, Chang Liu, Zetian Jiang, Xiaokang Yang, and Junchi Yan. Pygmtools: A python graph matching toolkit. *Journal of Machine Learning Research*, 25(33):1–7, 2024. URL https://jmlr.org/papers/v25/23-0572.html.

Andrei Zanfir and Cristian Sminchisescu. Deep learning of graph matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2684–2693, 2018. doi: 10.1109/CVPR.2018.00284.

Botong Zhang, Shuo Li, and Osbert Bastani. Conformal structured prediction. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=2ATD8a8P3C.

A FROZEN COMMON ON SPAIR-71K

Protocol and score interface. We use the released VGG16 COMMON checkpoint for SPair-71k (Lin et al., 2023; Min et al., 2019) without parameter updates. The official test annotations contain both directions of each image pair; retaining one canonical direction gives 6,117 unique pairs. We use the benchmark's intersection filter, which keeps keypoints present in both images and yields square complete-permutation problems with $n \in [3, 20]$. Thus this experiment evaluates the paper's complete-matching setting rather than partial correspondence with outliers.

The synthetic experiments form P by Sinkhorn-normalizing generated scores and set $W = \log P$. COMMON instead returns unnormalized neural compatibilities. A forward hook extracts its final unary affinity $A(x)$ immediately before Hungarian decoding; we take $W = A$ and $s_{ij}(x) = -A_{ij}(x)$. We apply no additional Sinkhorn transformation. In both protocols the remaining construction is identical:

$$S(x, \pi) = \max_i s_{i,\pi(i)}(x), \qquad E_\alpha(x) = \{(i, j) : s_{ij}(x) \le q_\alpha\}.$$

Neural inference is cached once. For each of five seeds, a uniform random split assigns 2,000 pairs to calibration and 4,117 to test; all three values of α reuse that split.

α	Coverage	Row	Exact	Feasible	Edges	Density	Support	Forced	Partners	Amb. P90	Counted	Set Med.
.05	95.24	85.64	68.29	99.08	43.57	73.39	42.85	0.66	5.06	12	91.53	216
.10	90.53	85.64	68.29	95.84	33.71	60.40	31.76	3.46	3.85	11	93.36	64
.20	80.24	85.64	68.29	86.13	23.64	45.27	19.79	14.26	2.48	8	95.03	12

Table 6: Frozen COMMON aggregate results across five splits. Coverage, Row, Exact, Feasible, Density, Forced, and Counted are percentages. Support counts edges extending to at least one perfect matching; Partners is the supported partner count per row, with empty graphs contributing zero. Amb. P90 is the 90th percentile largest ambiguous supported component. Component permanents are counted exactly up to 12 rows; Counted is the fraction of feasible graphs fully counted and Set Med. is conditional on that subset.

A.1 POOLED VERSUS SIZE-AWARE CALIBRATION

The pooled $\alpha = .10$ target holds marginally but not by size: small problems overcover, while several well-populated $n = 6$–13 groups undercover. This is the behavior anticipated by Remark 1, compounded by category and difficulty composition. Because n is observed before prediction, we also apply split conformal separately within prespecified size groups. For a group g, we compute the usual conformal rank and cutoff $q_{\alpha,g}$ using only calibration examples with $g(n_k) = g$, then return $\mathcal{G}_{q_{\alpha,g(n)}}(x)$. Applying Proposition 1 within each group gives coverage conditional on that group under within-group exchangeability; it does not give instance-conditional coverage.

We use one group for each $n = 3, \ldots, 16$ and pool the 19 rare examples with $n = 17, \ldots, 20$ into one prespecified tail group. Figure 1 compares this size-aware construction with the pooled cutoff on the same five splits. Size-aware calibration moves the common sizes toward the 90% target by reallocating graph size: previously undercovered sizes retain more edges, while overcovered sizes can retain fewer. Table 7 gives representative operating points. Overall coverage and mean density barely change, but the redistribution toward larger problems raises the mean retained-edge count from 33.71 to 37.73 and the mean feasible-partner count from 3.85 to 4.18.

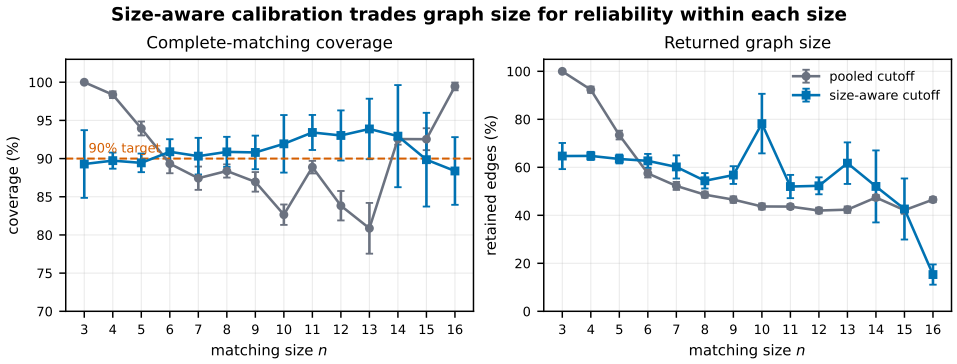

Figure 1: Pooled versus size-aware SPair-71k calibration at $\alpha = .10$. Curves show means across five splits and bars show one standard deviation. The left panel reports complete-matching coverage; the right reports retained-edge density. Size-aware calibration uses one cutoff per n for $n \leq 16$. The rare $n = 17$–20 tail group is omitted from the plot.

	Size cal.	Coverage		Density	
n	count	Pooled	Size-aware	Pooled	Size-aware
3	157	100.0	89.3	100.0	64.7
6	234	89.3	90.9	57.5	62.7
10	161	82.7	91.9	43.7	78.2
13	51	80.9	93.9	42.4	61.7
16	59	99.4	88.4	46.5	15.3
Overall	–	90.5	90.7	60.4	60.0

Table 7: Representative pooled and size-aware SPair results at $\alpha = .10$, in percent. Size cal. count is the mean number of calibration examples in the exact-n group; entries are means across five splits.

The calibration cost becomes explicit in the rare tail: its calibration groups contain only four to nine examples, so in three of five splits the conformal rank exceeds the group size, giving $q_{\alpha,g} = +\infty$ and a complete graph. Coarser bins would reduce this cost but would guarantee coverage only conditional on the coarser bin. Thus size stratification mitigates pooled size dependence rather than providing free conditional coverage.

A.2 Qualitative Feasible-Partner Queries

Figure 2 visualizes the local question answered by $\mathcal{F}_{i,\alpha}(x)$. The bird example contains five forced singleton components and one ambiguous three-row component: the green query certifies a correspondence that is fixed even though the complete prediction set contains four permutations, while the orange query exposes alternatives confined to the three-row group. In the harder sheep example, all 17 rows lie in one ambiguous component, so the orange query is globally coupled to the rest of the assignment. Every highlighted query edge extends to at least one complete one-to-one matching.

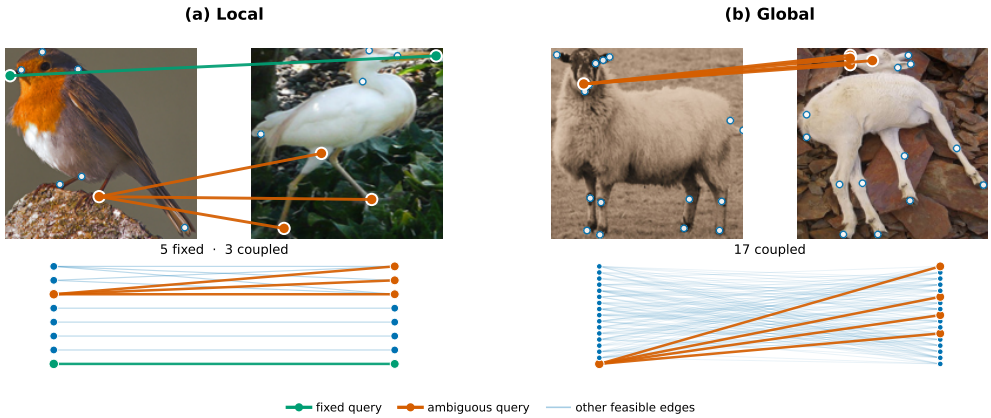

Figure 2: Feasible-partner queries for frozen COMMON on SPair-71k at $\alpha = .10$ (split seed 101). Green marks a forced query, orange an ambiguous query, and pale blue the remaining feasible edges. Left: five correspondences are fixed and ambiguity is confined to a three-row component. Right: one component couples all 17 rows.

Annotated difficulty. At $\alpha = .10$, viewpoint gives the clearest uncertainty response: from level 0 to 2, row accuracy falls from 89.4% to 78.9% while density rises from 56.8% to 68.7%. Scale and truncation also have denser graphs at higher annotation levels, whereas occlusion is non-monotone. These are descriptive associations: difficulty, category, and n are confounded, so Table 8 does not establish conditional calibration or a causal effect.

Annotation	Level	Coverage	Row	Density
Viewpoint	0	91.2	89.4	56.8
	1	89.4	81.3	64.1
	2	90.6	78.9	68.7
Scale	0	90.4	85.4	58.9
	1	89.7	85.3	60.3
	2	92.6	87.2	65.7
Truncation	0	90.3	85.5	54.7
	1	90.6	87.0	66.6
	2	90.3	83.4	68.0
	3	93.4	88.7	76.4
Occlusion	0	90.5	85.7	59.2
	1	91.4	86.9	63.6
	2	89.8	84.2	60.8
	3	90.0	84.2	66.6

Table 8: SPair-71k difficulty analysis at $\alpha = .10$, pooled over five splits. All entries except the ordinal level are percentages.

B Additional Computational Comparisons

B.1 Output-Sensitive Explicit Lists

Scores and calibration. Let $c_{ij}(x) = -W_{ij}(x)$ and

$$L(x, \pi) = \sum_i c_{i,\pi(i)}(x), \qquad L^\star(x) = \min_{\sigma \in S_n} L(x, \sigma).$$

The scalable additive predictor calibrates $S_{\mathrm{reg}}(x, \pi) = L(x, \pi) - L^\star(x)$, then returns every assignment satisfying $L(x, \pi) \le L^\star(x) + q_{\mathrm{reg}}$. This differs from the raw `sum_nll` score in the small-n

experiment by an input-dependent offset, so we report it separately as additive regret. The rank predictor uses the strict rank

$$R(x, \pi^\star) = 1 + \#\{\pi : L(x, \pi) < L(x, \pi^\star)\}$$

and conformally calibrates its top-K prefix. Boundary cost ties are included.

Exact sequential generation. Both explicit predictors use a Lawler–Murty partition of assignment space (Murty, 1968). Each heap node fixes a row prefix, forbids selected edges, and calls SciPy's linear assignment solver; the same solver underlies constrained graph comparisons. Popping nodes therefore produces complete assignments in nondecreasing additive cost without visiting all $n!$ assignments. Additive generation stops when the next cost exceeds $L^\star + q_{\mathrm{reg}}$; rank generation stops after K and any boundary ties. Tests compare the generated order with exhaustive sorting for $n = 2, \ldots, 7$, with separate checks for thresholds, caps, and forbidden edges.

SPair predictor	Coverage	Size med.	P90	P99	Maximum	Build (ms)	Support (ms)
Graph (edges)	90.53	25	64	124	179	0.004	0.244
Additive-regret list	90.23	2	18	105	1,401	0.521	1.147
Rank-K list	90.53	10–11	11	11	11	1.750	0.024

Table 9: Full SPair explicit-list comparison at $\alpha = .10$, pooled over five splits. Size counts edges for the graph and complete assignments for a list. Build materializes the representation; Support extracts every feasible row–column pair, by allowed-edge extraction for the graph, constrained assignment solves for additive regret, and a scan of the rank list.

Additive-regret and rank-K materialization completed below the 100,000-output cap for all 20,585 SPair test cases. During calibration, 55–67 of 2,000 ranks per split exceeded the rank cap of 1,000, but the calibrated K was only 10 or 11; hence the relevant order statistic was exact. Median ideal payloads were 7 bytes for bit-packed graph adjacency (50 bytes for sparse endpoints), 16 bytes for the additive list, and 77 bytes for rank-K. These payloads use the minimum integer width for column indices and exclude Python containers and the search queue.

Sorting predictor	Coverage	Size med.	P90	Count status	Construction	Ideal payload
Graph (edges)	90.04	80	80	exact	0.003 ms	50 / 160 B
Additive-regret list	89.84	6.88M	7.96M	99.48% exact	first 10k: 1.879 s	138 MB
Rank-K list	–	$\geq$ 5.61M	–	lower bound	–	$\geq$ 112 MB

Table 10: Block-sorting comparison at $n = 20$, $\alpha = .10$, and eight seeds. Graph payload is bit-packed adjacency/sparse endpoints; list payload stores one-byte column indices and excludes implementation overhead. Additive construction was timed on ten fixed test instances per seed; none completed before the 10,000-output cap.

Sorting uses 2,000 calibration and 3,000 test examples per seed. Factoring the five size-four blocks gives exact additive counts whenever a constrained solve certifies that no accepted cross-block assignment exists; this holds for 99.48% of test cases, and the remaining reported counts are lower bounds. Across all cases, 99.0% of additive sets exceed one million assignments. The analogous block-preserving calibration gives per-seed rank-K lower bounds from 5,580,648 to 5,884,053 (median 5,608,149); omitted cross-block assignments can only increase the global rank, so coverage is not reported for this unmaterialized lower-bound predictor. Median graph support extraction is 0.345 ms. Runtime constants depend on the particular exact M-best implementation, whereas the assignment counts and certified lower bounds do not.

B.2 Feasible-Partner Scaling

The Support operation in Table 2 first finds one perfect matching and then identifies all allowed edges through strongly connected components of the alternating row graph. Table 11 gives the density sensitivity under the same isolated-process protocol. Every run completed.

	Support time (ms)			
n	10%	50%	90%	50% peak MiB
64	0.296	0.400	0.378	59.1
256	1.45	3.48	3.84	61.7
1,024	18.18	54.43	51.99	105.9
2,048	78.05	231.37	292.14	245.3

Table 11: Feasible-partner extraction versus retained-edge density. Entries are medians across three isolated-process replicates, each summarizing ten matrices after warm-up. Peak memory includes Python, the input, and the retained support matrix.

C Additional Controlled Real-Data Experiments

C.1 WillowObject

Using `pygmtools`, we Sinkhorn-normalize a geometry score for labeled WillowObject pairs. Five splits use 100 calibration and 200 test pairs; every pair has $n = 10$.

Score	m	α	Target	Coverage	Point	Set Med.	Edges
`max_edge`	100	.05	95.05	93.4 ± 1.5	91.1 ± 1.8	5.2 ± 4.8	16.3 ± 4.2
`max_edge`	100	.10	90.10	88.4 ± 2.4	91.1 ± 1.8	1.0	10.9 ± 0.3
`max_edge`	100	.20	80.20	79.6 ± 4.0	91.1 ± 1.8	1.0	10.1 ± 0.2
`worst_rank`	100	.05	95.05	95.7 ± 1.8	91.1 ± 1.8	46.4 ± 35.1	26.0 ± 5.5
`worst_rank`	100	.10	90.10	91.8 ± 3.3	91.1 ± 1.8	5.8 ± 3.2	18.0 ± 4.5
`worst_rank`	100	.20	80.20	85.6 ± 1.6	91.1 ± 1.8	1.0	10.0

Table 12: WillowObject over five splits. Target is $r_\alpha/(m+1)$; Point is exact-permutation accuracy.

At $\alpha = .10$, `max_edge` coverage is 88.4%, the median set has one permutation, 95.1% of rows are forced, and the ambiguous-component P90 is two rows. Empty-set frequencies at $\alpha = .05, .10, .20$ are $1.3\%, 7.9\%, 18.5\%$; the rank graph is consistently larger and overcovers.

C.2 PascalVOC Base-Matcher Sensitivity

To separate the conformal wrapper from the score model, we evaluate geometry-only, linear, and two-layer MLP pair scorers on PascalVOC pairs with at most nine matched keypoints. Each learned scorer is trained before calibration and evaluated on held-out pairs from the same sampler. Runs are not paired instance-by-instance, so Table 13 is a sensitivity analysis rather than a ranking of matching architectures.

Scorer	Coverage	Point	Set Med.	Edges	Forced	Amb. P90
Geometry	88.7	48.3	342	34.6	2.8	9
Linear	88.8	47.3	48	24.1	7.3	9
MLP	88.2	44.2	50	25.2	7.4	9

Table 13: PascalVOC sensitivity at $\alpha = 0.10$. Coverage, point accuracy, and forced-row fraction are percentages; Amb. P90 is the 90th percentile largest ambiguous component size. Geometry uses 10 repeated splits and 5,000 test instances; learned scorers use held-out pairs from 10 independently trained runs with 5,000 test instances.

Coverage remains near target, but representation efficiency changes substantially with the score model. The learned scorers reduce the median set from 342 to about 50 and retain fewer edges, even though point accuracy does not improve in this lightweight study. They do not make ambiguity fully local: the largest ambiguous component can still span all nine rows. This supports a qualified conclusion: the conformal graph always represents and measures uncertainty, whereas localization depends on the base matcher's score geometry.

D SPLIT-CONFORMAL MATCHING-GRAPH ALGORITHM

Algorithm 1 shows how calibration selects one member of the nested family of exactly graph-represented prediction sets.

Algorithm 1 Split-conformal matching graph

Require: Fixed edge nonconformity s, calibration data $\{(x_k, \pi_k)\}_{k=1}^{m}$, level α, test input x

1: **for** $k = 1, \dots, m$ **do**
2: $\quad S_k \leftarrow \max_i s_{i,\pi_k(i)}(x_k)$
3: Sort $S_{(1)} \le \cdots \le S_{(m)}$; $r \leftarrow \lceil (m+1)(1-\alpha) \rceil$
4: $q_\alpha \leftarrow S_{(r)}$ if $r \le m$, else $+\infty$
5: $A_{ij}(x) \leftarrow \mathbf{1}\{s_{ij}(x) \le q_\alpha\}$ for all i, j
6: Form $\mathcal{G}_\alpha(x)$ from the edges with $A_{ij}(x) = 1$
7: **return** $\mathcal{G}_\alpha(x)$; optionally report its connected components

E ADDITIONAL LIMITATIONS AND SCOPE

Statistical interpretation. The pooled guarantee is marginal over the calibration sample and a new test example; it does not provide instance-conditional or general subgroup-conditional coverage. The size-stratified SPair experiment gives coverage conditional on its prespecified size groups under within-group exchangeability, not conditional on category, difficulty, or an individual image pair. Both constructions rely on keeping the matcher fixed during calibration and testing. Distribution shift, changes to the object-pair sampler, or post-calibration model updates can therefore invalidate the rank argument. The SPair splits estimate coverage for a uniformly randomized official test-pair pool; because pairs may share images, evaluation on entirely new images would require an image-grouped protocol. Applications requiring reliability by difficulty or under distribution shift would need additional stratified, weighted, or shift-aware calibration assumptions.

Representation efficiency versus cardinality. Poorly ranked edge scores can produce a dense, globally connected graph. This is not a coverage failure, but it may make the uncertainty summary less informative. Conversely, a sparse representation need not contain few permutations: the sorting experiment exhibits trillions of accepted permutations encoded by roughly one hundred edges. Atomic cardinality, storage, query time, and localization are distinct efficiency criteria. Counting perfect matchings remains #P-hard (Valiant, 1979), so the graph is normally queried rather than converted back into a list. SPair cardinalities are reported only when every ambiguous component has at most 12 rows; the reported median is therefore conditional on the countable feasible subset.

The bottleneck score is appropriate when every ground-truth edge must be retained, but it can be conservative when a task tolerates a few local errors. Removing one graph edge excludes every permutation containing it, so small-n adaptive-mass and top-k baselines can return fewer atomic outputs. The proposed advantage is an exact matching-native representation and query interface, not volume optimality.

Experimental scope. The `adaptive_mass` baseline is an exact atomic-label instantiation of the CSP framework, not an approximation or port of the task-specific DAG optimizers in Zhang et al. (2025); exhaustive adaptive-mass comparisons stop at $n \le 9$. Exact output-sensitive Lawler–Murty comparisons extend to cached SPair scores, while the hard sorting regime uses exact structural counts or certified lower bounds together with a fixed materialization cap. These timings exclude neural inference. WillowObject and PascalVOC use controlled lightweight scorers, whereas SPair uses a released pretrained COMMON checkpoint; we did not find a compatible released COMMON+ checkpoint. The SPair intersection filter removes unmatched keypoints and therefore does not test partial matching with outliers. Finally, the component product is a factorization of the accepted *set*, not posterior independence between correspondence decisions.

F Deferred Proofs and Remarks on Graph Representability

Proof of Proposition 2. Every $\pi \in \mathcal{C}$ uses only edges in $\mathrm{supp}(\mathcal{C})$, giving $\mathcal{C} \subseteq \mathcal{M}(\mathrm{supp}(\mathcal{C}))$. Equality immediately implies representability. Conversely, if $\mathcal{C} = \mathcal{M}(E)$, every edge used by a member of $\mathcal{C}$ lies in E. Hence $\mathrm{supp}(\mathcal{C}) \subseteq E$ and $\mathcal{M}(\mathrm{supp}(\mathcal{C})) \subseteq \mathcal{M}(E) = \mathcal{C}$. The two inclusions give equality. Any representing edge set must contain $\mathrm{supp}(\mathcal{C})$, proving minimality. □

Proof of Theorem 1. If $S(\pi) = \max_i s(i, \pi(i))$, then $S(\pi) \le t$ if and only if every $s(i, \pi(i)) \le t$, so $\mathcal{C}_t = \mathcal{M}(E_t)$ for $E_t = \{(i, j) : s(i, j) \le t\}$.

Conversely, suppose every threshold set is graph-representable. Fix π and let $t = \max_i s^\star(i, \pi(i))$. Using π as a competitor gives $t \le S(\pi)$. For every i, choose a minimizer σ_i defining $s^\star(i, \pi(i))$. Since $S(\sigma_i) \le t$, the edge $(i, \pi(i))$ belongs to $\mathrm{supp}(\mathcal{C}_t)$. Thus $\pi \in \mathcal{M}(\mathrm{supp}(\mathcal{C}_t)) = \mathcal{C}_t$ by Proposition 2, yielding $S(\pi) \le t$. Therefore $S(\pi) = \max_i s^\star(i, \pi(i))$. □

Representable sets are coarse. If $\mathcal{C} \subsetneq S_n$ is graph-representable, then

$$|S_n \setminus \mathcal{C}| \ge (n-1)!.$$

Indeed, by Proposition 2, $\mathcal{C} = \mathcal{M}(\mathrm{supp}(\mathcal{C}))$. If $\mathrm{supp}(\mathcal{C})$ were the complete edge set, then $\mathcal{C} = \mathcal{M}(\{1, \dots, n\}^2) = S_n$, contradicting $\mathcal{C} \subsetneq S_n$. Hence some edge $(i, j) \notin \mathrm{supp}(\mathcal{C})$, and every one of the $(n-1)!$ permutations satisfying $\pi(i) = j$ is excluded.

Injective scores have nonrepresentable threshold sets. Let $n \ge 3$ and let S be injective on S_n. Let $\pi^\dagger = \arg\max_\pi S(\pi)$, which is unique, and let t be the second-largest value of S. Then

$$\mathcal{C}_t = S_n \setminus \{\pi^\dagger\}.$$

Its complement contains only one permutation, whereas any proper graph-representable subset of S_n must exclude at least $(n-1)! \ge 2$ permutations by the observation above. Hence $\mathcal{C}_t$ is not graph-representable. Equivalently, S is not a bottleneck score. Moreover, a bottleneck score can take at most n^2 distinct values on S_n, since $\max_i s(i, \pi(i))$ must equal one of the n^2 edge scores.

Proof of Proposition 3. The additive score $S_{\mathrm{sum}}(\pi) = \sum_i s(i, \pi(i))$ fails to be injective only if $\sum_i s(i, \pi(i)) = \sum_i s(i, \sigma(i))$ for some $\pi \ne \sigma$. Each such equation is a nontrivial linear constraint on s, because the permutation matrices of π and σ are distinct, and therefore defines a hyperplane of Lebesgue measure zero in $\mathbb{R}^{n \times n}$; the union over the finitely many pairs $\pi \ne \sigma$ is again null. For every s outside this null set S_{sum} is injective, so the injectivity argument above applies. The regret score S_{reg} differs from S_{sum} by a constant that does not depend on π, so it is injective under the same condition and its threshold sets are those of S_{sum} at shifted thresholds. The same conclusion therefore holds for regret. □

A minimal witness. The obstruction is visible at $n = 3$. Take $s(1, 1) = s(2, 2) = 1$ and $s(i, j) = 0$ for all other edges. Under the additive score the identity is the unique permutation with $S_{\mathrm{sum}} = 2$, and every other permutation scores 0 or 1, so at $t = 1$ we have $\mathcal{C}_1 = S_n \setminus \{\mathrm{id}\}$. Each of the nine edges is used by exactly two permutations, so at least one member of $\mathcal{C}_1$ uses it; hence $\mathrm{supp}(\mathcal{C}_1)$ is the complete edge set and $\mathcal{M}(\mathrm{supp}(\mathcal{C}_1)) = S_n \ne \mathcal{C}_1$. By Proposition 2, $\mathcal{C}_1$ is not graph-representable. The mechanism is transparent: the edges $(1, 1)$ and $(2, 2)$ are individually acceptable but jointly are not, and a bipartite graph has no way to forbid their conjunction while admitting each separately. Because the strict inequalities defining this threshold set are preserved under sufficiently small perturbations of every entry of s, the counterexample is not an artifact of ties.

Which thresholds fail. A score that is not a bottleneck score may still be representable at isolated thresholds. For instance $\mathcal{C}_t = S_n$ whenever t exceeds the maximum score, and $\mathcal{C}_t = \emptyset$ whenever t is below the minimum; both are trivially representable. The injectivity argument above shows that the failure occurs at thresholds that exclude some but not most of the label space, which are precisely the operating points at which a useful conformal set sits. Representability at the single calibrated threshold q_α is therefore not a meaningful weakening of Definition 1: q_α is determined by the calibration sample and α is chosen by the user, so neither is available at design time.

Conservative graph relaxation. When a threshold set is not graph-representable, it may still be enlarged to its graph closure $\mathcal{M}(\mathrm{supp}(\mathcal{C}_\alpha(x)))$. By Proposition 2 this is a superset of $\mathcal{C}_\alpha(x)$, so it inherits the coverage guarantee of the underlying score and, once constructed, supports the same membership, feasibility, representative-prediction, and component queries as the exact construction. The relaxation is not exact: it can contain permutations rejected by the score, so its size no longer reflects the calibrated threshold set. Moreover, computing the support of a general structured set may itself require difficult constrained searches; this closure is a representational observation rather than a generic efficient algorithm. Combining Proposition 2 with Theorem 1, the relaxation is exact at every threshold if and only if the score has a bottleneck form. Finally, for $n \leq 2$ every subset of S_n is graph-representable, so the phenomenon begins at $n \geq 3$.

G Calibration Variability and Experimental Reference Errors

The coverage proposition requires only exchangeability, but interpreting variation across repeated calibration draws uses stronger assumptions. Suppose the scores are i.i.d. with continuous CDF F and $r_\alpha \leq m$. Conditional on the calibration data, test coverage is

$$P_{\mathrm{cal}} = F(S_{(r_\alpha)}).$$

The probability integral transform and the order-statistic law give

$$P_{\mathrm{cal}} \sim \mathrm{Beta}(r_\alpha, m + 1 - r_\alpha).$$

This Beta law does not follow from exchangeability alone, and conditional on a fixed calibration draw the coverage probability is a fixed number. If a test sample contains N independent examples, its empirical coverage has additional conditional binomial variance $P_{\mathrm{cal}}(1 - P_{\mathrm{cal}})/N$. For independent repeated splits, the reference standard error of their mean combines the Beta calibration variance with the expected binomial variance. The experiment captions use this calculation only to contextualize Monte Carlo variation; it is not part of the coverage guarantee. If $r_\alpha = m + 1$, then $q_\alpha = +\infty$ and coverage is one. Ties can instead make the conformal threshold conservative.

H Deferred Proof of Split Conformal Coverage

Proof of Proposition 1. The scores

$$S(X_1, \Pi_1), \ldots, S(X_m, \Pi_m), S(X_{m+1}, \Pi_{m+1})$$

are exchangeable because they are deterministic functions of exchangeable examples and a fixed matcher that is independent of the calibration data. The standard split conformal rank argument gives

$$\Pr\{S(X_{m+1}, \Pi_{m+1}) \leq q_\alpha\} \geq 1 - \alpha.$$

By construction,

$$\Pi_{m+1} \in \mathcal{C}_\alpha(X_{m+1}) \quad \Longleftrightarrow \quad S(X_{m+1}, \Pi_{m+1}) \leq q_\alpha.$$

Combining these two statements proves the claim. □