Exact Combinatorial Optimization for Partial Permutation Synchronization

Mohammad Mahdi Omati
Department of Electrical Engineering
Sharif University of Technology
mohammad_omati@yahoo.com

Arash Amini
Department of Electrical Engineering
Sharif University of Technology
aamini@sharif.edu

Abstract

In permutation synchronization, the goal is to recover cycle-consistent correspondences among the keypoints of a collection of images from noisy pairwise matchings. In this work, we keep the problem in its original combinatorial form throughout, unlike spectral and continuous methods that embed permutations into an orthogonal space, losing tightness, and then round the solution back. We show that, once the assignments of all other images are fixed, the objective is exactly linear in the assignment of the remaining image, because the feasibility constraints alone remove every interaction between its keypoints. Each update is therefore a rectangular linear assignment problem whose constraint matrix is totally unimodular, so it is solved to global optimality with an integral solution and without any rounding. We prove that the resulting block-coordinate ascent increases the objective monotonically, terminates after finitely many updates, and reaches a block-wise optimum when ties are resolved in favour of the current assignment, and we derive computable upper bounds on the optimal objective. On the multi-view sequences of the ETH3D benchmark, with ground truth derived from calibrated 3D geometry, our method preserves exact cycle consistency and outperforms state-of-the-art synchronization and multi-graph matching methods in both F-score and precision.

1 Introduction

Establishing correspondences between visual observations is a fundamental task in pattern recognition and computer vision, with downstream impact on shape deformation modelling [1, 2], object tracking [3, 4], 3D reconstruction from imagery [5, 6], graph matching [7, 8], and image registration [9, 10]. Even for a single pair of images, matching keypoints while respecting their geometric relations is a quadratic assignment problem, one of the hardest problems of combinatorial optimization [11]. Real-world pipelines, moreover, need to match many images at once. In this multi-object setting every pairwise match is linked to the others through transitivity, so an error in one match propagates through the collection, and a solver that treats each pair separately cannot correct it [12–14].

Permutation synchronization addresses this difficulty by exploiting the over-determined structure of a multi-image collection [15, 16]. Its core requirement is cycle consistency, under which composing the matchings around any closed path of images yields the identity. This requirement has proven useful across multi-alignment [17–19], multi-shape analysis [20–22], multi-image matching [23–27], and multi-graph matching [28]. Beyond correspondence problems, synchronization has also been studied for broader classes of transformations [29–33]. By aggregating the pairwise matchings into a single coherent assignment over the whole collection, synchronization acts as a denoising step, and it returns matchings that respect transitivity throughout, which no pairwise matcher guarantees.

The standard formulation of synchronization is a trace maximization over permutation matrices subject to integrality and row- and column-sum constraints. Spectral approaches replace this program

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

by a continuous one over the orthogonal group, and they presume full bijective correspondences, so they cannot express a partial match [16, 34]. A non-iterative spectral scheme admits partial matches but does not enforce cycle consistency, so its pairwise outputs can contradict one another although true matchings never do [35]. Later methods that admit partial matches still work on a continuous version of the problem. A scheme based on non-negative matrix factorization computes a continuous solution and has to project it back at the end, so its output is determined by a step that the optimized criterion does not take into account [36]. The Stiefel-manifold formulation replaces the binary constraints by an orthogonality constraint, a relaxation (in the sense of losing tightness), and then promotes sparsity to push the solution towards binary values [37]. Because the problem it solves is this looser one and not the original, its sparsity is only approximate, and the sparsity term can guarantee neither that the result satisfies the combinatorial constraints nor that it is an optimal solution of the original problem. Other methods score each observed match by its consistency with the rest of the collection and obtain a matching only after thresholding or projecting these scores, so the final matching is again decided by a step outside the optimization [38, 39]. A projected power iteration has to project each iterate back onto the feasible set, so a rounding step sits inside every iteration, and its monotonicity concerns the projected sequence, not the combinatorial objective at a feasible point [25].

A second route leaves the continuous embedding and searches over matchings directly. A construction-then-refinement pipeline pairs a sequential or a parallel construction with a graph-matching local search [8], and GREEDA carries the same recipe to large, sparse and incomplete instances behind an added swap-based refinement [40]. None of these solvers carries a convergence guarantee. Each improves a heuristic objective by trial exchanges with no formal assurance of reaching even a local optimum of the synchronization criterion, so the returned matching depends on where the exchanges happen to terminate, and combining several local modules does not guarantee that the result reaches an acceptable level of accuracy.

This paper works on the combinatorial problem itself, one image at a time. Our key observation is that, once the assignments of all other images are fixed, the constraints on the remaining image make the objective exactly linear in its assignment, for any observed matchings. Each update is therefore a linear assignment problem that is solved exactly, and every iterate is a partial permutation whose pairwise matchings are cycle-consistent by construction.

Our contribution. The main contributions of this work are summarized as follows.

- **A direct combinatorial approach.** We work directly in the space of binary partial permutation matrices and never pass through a relaxation of the problem that loses tightness. Every iterate is a valid partial permutation, and the pairwise matchings read off from it are cycle-consistent by construction.
- **An exactly linear objective for each image.** Feasibility alone requires the keypoints of one image to occupy distinct universe labels, and this requirement removes every interaction between two keypoints of the same image, for any observed matchings. With the other images fixed, the objective is therefore exactly linear in the assignment of the remaining image, although as a whole it is an indefinite quadratic.
- **Each update solved exactly, by assignment and without rounding.** The constraint matrix of each per-image update is totally unimodular, so the linear program of that update has an integral optimum, which a rectangular linear assignment solver computes exactly. Because the feasible set is a product over images, updating one image never affects the feasibility of the others. No thresholding, no projection and no rounding is performed at any point.
- **Guarantees and computable bounds.** The objective increases monotonically, and it is integer-valued on binary observations, which bounds the number of improving updates and yields finite termination, with a block-wise optimal limit when ties are resolved in favour of the current assignment. We further derive computable upper bounds on the optimal objective, against which any solution can be measured.
- **Evaluation against geometric ground truth.** We evaluate on the multi-view sequences of the ETH3D benchmark, with ground truth derived from calibrated 3D geometry and noisy matchings produced by a real feature matcher. Our method outperforms state-of-the-art and benchmark methods in both F-score and precision, while remaining among the fastest of the methods compared.

Notation. Bold upper-case letters denote matrices and bold lower-case letters vectors. $\mathbb{1}_p$ is the all-ones vector of length p, $\mathbf{I}_p$ the $p \times p$ identity, and vector inequalities hold entrywise. For matrices of equal size, $\langle \mathbf{A}, \mathbf{B} \rangle = \operatorname{tr}(\mathbf{A}^T\mathbf{B})$ is the Frobenius inner product and $\|\mathbf{A}\|_F$ the Frobenius norm. $\operatorname{Diag}(\mathbf{a})$ is the diagonal matrix with diagonal $\mathbf{a}$, and $\operatorname{diag}(\mathbf{A})$ is the vector of diagonal entries of $\mathbf{A}$. $\operatorname{vec}(\cdot)$ stacks the columns of a matrix, $\otimes$ is the Kronecker product, and $[x]_+ = \max(x, 0)$. For a symmetric matrix $\mathbf{W}$, $\lambda_1(\mathbf{W}) \geq \lambda_2(\mathbf{W}) \geq \cdots$ are its eigenvalues in non-increasing order. Throughout, k is the number of images, m_i the number of keypoints of image i, $m = \sum_i m_i$, and d the number of universe labels.

2 Problem Formulation

We consider a collection of k images, where image i contains m_i keypoints. For each pair (i, j), the correspondence between the keypoints of image i and those of image j is represented by a partial permutation matrix

$$\mathbf{P}_{ij} \in \mathbb{P}_{m_i m_j} := \big\{ \mathbf{X} \in \{0,1\}^{m_i \times m_j} \,:\, \mathbf{X}\mathbb{1}_{m_j} \leq \mathbb{1}_{m_i},\ \mathbf{X}^T\mathbb{1}_{m_i} \leq \mathbb{1}_{m_j} \big\}.$$

When these matrices are full bijections, the family $\mathcal{P} = \{\mathbf{P}_{ij}\}_{i,j=1}^k$ is cycle-consistent if $\mathbf{P}_{i\ell}\mathbf{P}_{\ell j} = \mathbf{P}_{ij}$ holds for every triplet (i, ℓ, j). A convenient way to parameterize all cycle-consistent families is through a universe of d labels, d being the number of distinct keypoint identities across the collection. We denote by

$$\overline{\mathbb{P}}_{m_i d} := \big\{ \mathbf{X} \in \mathbb{P}_{m_i d} \,:\, \mathbb{1}_{m_i}^T \mathbf{X} \mathbb{1}_d = \min(m_i, d) \big\}$$

the maximal partial permutations, namely those that label as many keypoints of image i as the two injectivity constraints permit. When $m_i \leq d$ this is the familiar equality $\mathbf{X}\mathbb{1}_d = \mathbb{1}_{m_i}$. When $m_i > d$, the partial regime of our experiments in which an image carries more keypoints than the universe has labels, exactly $m_i - d$ keypoints remain unlabelled, and imposing $\mathbf{X}\mathbb{1}_d = \mathbb{1}_{m_i}$ instead would leave the feasible set empty. Cycle consistency among the pairwise matrices holds if and only if there exist object-to-universe assignments $\mathbf{P}_i \in \overline{\mathbb{P}}_{m_i d}$ with $\mathbf{P}_{ij} = \mathbf{P}_i\mathbf{P}_j^T$ for all i, j, an equivalence that remains valid for partial maps [24, 36].

Given the noisy observed family $\mathcal{P}$, permutation synchronization seeks the assignments that best fit the observations, $\arg\max \sum_{i,j} \operatorname{tr}(\mathbf{P}_{ij}^T \mathbf{P}_i \mathbf{P}_j^T)$ over $\mathbf{P}_i \in \overline{\mathbb{P}}_{m_i d}$ [16]. Stacking the assignments and the observations as

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_1^T & \cdots & \mathbf{P}_k^T \end{bmatrix}^T \in \mathbb{R}^{m \times d}, \qquad \mathbf{W} := [\mathbf{P}_{ij}]_{i,j=1}^k \in \mathbb{R}^{m \times m}, \qquad \mathbb{U} := \overline{\mathbb{P}}_{m_1 d} \times \cdots \times \overline{\mathbb{P}}_{m_k d},$$

the problem reads

$$\boxed{\begin{aligned} \max_{\mathbf{P} \in \{0,1\}^{m \times d}} \quad & f(\mathbf{P}) := \operatorname{tr}\big(\mathbf{P}^T\mathbf{W}\mathbf{P}\big) \\ \text{s.t.} \quad & \mathbf{P}_i\mathbb{1}_d \leq \mathbb{1}_{m_i}, \qquad \mathbf{P}_i^T\mathbb{1}_{m_i} \leq \mathbb{1}_d, \\ & \mathbb{1}_{m_i}^T\mathbf{P}_i\mathbb{1}_d = \min(m_i, d), \qquad i = 1, \ldots, k. \end{aligned}} \tag{1}$$

Equivalently $\mathbf{P} \in \mathbb{U}$, and since every constraint is stated per block, the feasible set is a product over images. Two features make (1) hard. First, $\mathbf{W}$ is indefinite in general, so the objective is a non-concave quadratic form. Second, the binary row- and column-sum constraints make the problem a combinatorial search that is NP-hard [11]. The method of the next section addresses both difficulties without leaving the permutation domain.

3 Exact Block-Coordinate Synchronization

Problem (1) is a quadratic program over a finite set. Our method rests on a single observation, namely that the difficulty arises only when all k images are considered jointly. Restricted to one image, the objective is exactly linear, and it can therefore be maximized exactly over the binary feasible set by a linear assignment solver. We establish both facts for a strictly larger family than (1), the quadratic-plus-linear objective

$$\max_{\mathbf{P} \in \mathbb{U}} \ F(\mathbf{P}) := \operatorname{tr}\big(\mathbf{P}^T\mathbf{W}\mathbf{P}\big) + \langle \mathbf{C}, \mathbf{P} \rangle, \tag{2}$$

with a symmetric $\mathbf{W} \in \mathbb{R}^{m \times m}$ and an arbitrary $\mathbf{C} \in \mathbb{R}^{m \times d}$, in which $\mathbf{C}$ carries unary evidence, such as an appearance prior on assigning a keypoint to a label or a price on assigning it at all. Problem (1) is the case $\mathbf{C} = \mathbf{0}$.

3.1 The per-image restriction is exactly linear

We regard F as a function of the assignment $\mathbf{P}_i$ of a single image, with all other blocks held fixed, and write $\mathbf{W}_{ij}$ for the (i, j) block of $\mathbf{W}$ and $\mathbf{C}_i$ for the rows of $\mathbf{C}$ that belong to image i.

Lemma 3.1 (Annihilation of the diagonal block)**.** *For every $\mathbf{P}_i \in \mathbb{P}_{m_i d}$, $\mathbf{P}_i \mathbf{P}_i^T = \mathrm{Diag}(\mathbf{P}_i \mathbb{1}_d)$, and consequently, for every symmetric $\mathbf{W}_{ii}$,*

$$\mathrm{tr}\big(\mathbf{P}_i^T \mathbf{W}_{ii} \mathbf{P}_i\big) = \big\langle \mathrm{diag}(\mathbf{W}_{ii}) \mathbb{1}_d^T,\ \mathbf{P}_i \big\rangle,$$

which is linear in $\mathbf{P}_i$ (for the proof, see Appendix A).

Every off-diagonal entry of $\mathbf{W}_{ii}$ is thus eliminated from the objective by the constraints alone, however dense or indefinite $\mathbf{W}_{ii}$ may be, since two keypoints of the same image never carry the same label. Only the coupling with the other images remains, and that coupling is linear in $\mathbf{P}_i$.

Proposition 1 (Exactness of the block subproblem)**.** *Let $\mathbf{W} = \mathbf{W}^T$ and $\mathbf{C}$ be arbitrary, fix every block except $\mathbf{P}_i$, and let $\mathbf{E}_i := \sum_{j \neq i} \mathbf{W}_{ij} \mathbf{P}_j$ be the cross-image evidence. Then for every $\mathbf{P}_i \in \mathbb{P}_{m_i d}$*

$$F(\mathbf{P}) = \big\langle \mathbf{V}_i,\ \mathbf{P}_i \big\rangle + \kappa_i, \qquad \mathbf{V}_i := 2\mathbf{E}_i + \mathrm{diag}(\mathbf{W}_{ii}) \mathbb{1}_d^T + \mathbf{C}_i, \tag{3}$$

where κ_i does not depend on $\mathbf{P}_i$ (for the proof, see Appendix A).

The restriction of F to one block is therefore exactly linear, for any symmetric $\mathbf{W}$ and any $\mathbf{C}$, with no assumption on the diagonal blocks and no requirement that the blocks be maximal. The same update thus applies to cycle-consistent multi-matching with arbitrary costs, and Appendix F compares it with multi-graph matching solvers on common objectives. A block step of this kind is available only because the feasible set is a product over images, so that one block may be changed alone while the iterate stays feasible. Orthogonality-constrained formulations do not share this structure, since the constraint $\mathbf{U}^T \mathbf{U} = \mathbf{I}_d$ on their continuous embedding $\mathbf{U} \in \mathbb{R}^{m \times d}$ ties every row of $\mathbf{U}$ together, and freezing the rows outside one image generally leaves no feasible choice for that image [37].

3.2 The subproblem is an exact linear assignment

Consider maximizing a linear objective $\langle \mathbf{C}, \mathbf{P} \rangle$, for an arbitrary cost matrix $\mathbf{C} \in \mathbb{R}^{m \times d}$, over the feasible set of (1). This is an integer linear program, and we show that its linear programming relaxation, in which the binary constraint is replaced by $\mathbf{0} \leq \mathbf{P} \leq \mathbb{1}\mathbb{1}^T$, is tight, so that no tightness is lost and the solution never leaves the permutation domain.

Definition 3.1 (Total unimodularity)**.** An integer matrix $\mathbf{D}$ is totally unimodular (TU) if every square submatrix of $\mathbf{D}$ has determinant in $\{-1, 0, 1\}$.

Theorem 3.2 (Hoffman–Kruskal integrality [41, 42])**.** *If $\mathbf{D}$ is totally unimodular and $\mathbf{b}$ is integral, then every vertex of the polyhedron $\{\mathbf{x} \geq \mathbf{0} : \mathbf{D}\mathbf{x} \leq \mathbf{b}\}$ is integral (for the proof, see Appendix A).*

To apply Theorem 3.2, we vectorize $\mathbf{P} \in \mathbb{R}^{m \times d}$ into the column-stacked vector $\mathbf{x} = \mathrm{vec}(\mathbf{P}) \in \mathbb{R}^{md}$. The requirement that each keypoint receive at most one label reads $(\mathbb{1}_d^T \otimes \mathbf{I}_m)\mathbf{x} \leq \mathbb{1}_m$. The requirement that each label be used at most once per image reads $(\mathbf{I}_d \otimes \mathbf{J})\mathbf{x} \leq \mathbb{1}_{kd}$, where $\mathbf{J} \in \{0, 1\}^{k \times m}$ encodes image membership, $J_{ir} = 1$ exactly when row r of $\mathbf{P}$ belongs to image i. Stacking the two gives

$$\mathbf{D} = \begin{bmatrix} \mathbb{1}_d^T \otimes \mathbf{I}_m \\ \mathbf{I}_d \otimes \mathbf{J} \end{bmatrix} \in \{0, 1\}^{(m+kd) \times md}, \qquad \mathbf{b} = \mathbb{1}_{m+kd}. \tag{4}$$

Lemma 3.3 (Total unimodularity of the constraint matrix)**.** *The matrix $\mathbf{D}$ in (4) is totally unimodular (for the proof, see Appendix A).*

Proposition 2 (Tightness of the linear programming relaxation)**.** *For every $\mathbf{C} \in \mathbb{R}^{m \times d}$, the linear program*

$$\max_{\mathbf{x} \in \mathbb{R}^{md}} \ \mathrm{vec}(\mathbf{C})^T \mathbf{x} \quad \textit{s.t.} \quad \mathbf{D}\mathbf{x} \leq \mathbf{b}, \quad \mathbf{x} \geq \mathbf{0}, \quad \mathbb{1}_{m_i}^T \mathbf{P}_i \mathbb{1}_d = \min(m_i, d),\ \ i = 1, \ldots, k, \tag{5}$$

attains its maximum at a point $\mathbf{x}^\star = \mathrm{vec}(\mathbf{P}^\star)$ with $\mathbf{P}^\star \in \mathbb{U}$. Moreover, (5) decouples across the blocks $\mathbf{P}_1, \ldots, \mathbf{P}_k$ into k independent rectangular linear assignment problems (for the proof, see Appendix A).

Algorithm 1 Exact block-coordinate permutation synchronization

Require: Observed matchings $\{\mathbf{P}_{ij}\}_{i,j=1}^{k}$, sizes $\{m_i\}$, universe size d, initial assignment $\mathbf{P}^{(0)} \in \mathbb{U}$
Ensure: Object-to-universe assignment $\mathbf{P} \in \mathbb{U}$

$\mathbf{W} \leftarrow [\mathbf{P}_{ij}]_{i,j=1}^{k}$ with $\mathbf{W}_{ii} = \mathbf{I}_{m_i}$, and $\mathbf{P} \leftarrow \mathbf{P}^{(0)}$
repeat
  for $i = 1, \dots, k$ **do**
    $\mathbf{E}_i \leftarrow \sum_{j \neq i} \mathbf{W}_{ij}\mathbf{P}_j$ ▷ cross-image evidence
    $\mathbf{P}_i \leftarrow \arg\max_{\mathbf{P}_i \in \overline{\mathbb{P}}_{m_i d}} \langle \mathbf{E}_i, \mathbf{P}_i \rangle$ ▷ exact rectangular LAP
  end for
until a full pass over the k images leaves $f(\mathbf{P})$ unchanged
return $\mathbf{P}$

Each of these rectangular linear assignment problems (LAPs) is solved to global optimality by the Jonker–Volgenant algorithm [43], and by Proposition 2 its optimizer is integral, so no projection, thresholding or rounding step is ever applied. In permutation synchronization, $\mathbf{C} = \mathbf{0}$ and $\mathbf{W}_{ii} = \mathbf{I}_{m_i}$, so the middle term of $\mathbf{V}_i$ is the constant matrix $\mathbb{1}_{m_i}\mathbb{1}_d^T$. Since every block in $\overline{\mathbb{P}}_{m_i d}$ carries exactly $\min(m_i, d)$ nonzeros, that term shifts the objective by a fixed amount, as does the positive factor 2, and neither changes the maximizer. The update of image i is therefore

$$\mathbf{P}_i \;\leftarrow\; \arg\max_{\mathbf{P}_i \in \overline{\mathbb{P}}_{m_i d}} \big\langle \mathbf{E}_i,\, \mathbf{P}_i \big\rangle, \qquad \mathbf{E}_i = \sum_{j \neq i} \mathbf{W}_{ij}\mathbf{P}_j. \tag{6}$$

Its meaning is a vote count. The entry $\mathbf{E}_i[r, c]$ is the number of other images containing a keypoint that $\mathbf{W}$ matches to keypoint r of image i and that currently carries label c, so the update gives every keypoint the label with the most support across the collection, exactly and jointly within the image. Algorithm 1 applies (6) cyclically over the images, and every iterate is an exact partial permutation.

The update is built on minorization–maximization principles [44]. The linear function $\langle \mathbf{V}_i, \mathbf{P}_i \rangle + \kappa_i$ minorizes the restriction of the objective to block i and touches it at the current iterate, as the two defining conditions require. By Proposition 1 it moreover coincides with that restriction everywhere, so the surrogate is tight and introduces no slack (for the proof, see Appendix A.6).

Guarantees, bounds and cost. Because each update maximizes the true restriction of the objective, f never decreases along the iterates. When $\mathbf{W}$ is binary, as it is for observed matchings, f is integer-valued, so every strictly improving update raises it by at least one. Since f is bounded above, the procedure terminates after finitely many passes. When the current assignment of an image is retained whenever it already attains the maximum of (6), the iterates become constant and their limit is block-wise optimal, a point at which no reassignment of a single image can increase the objective. We further derive two computable upper bounds on the optimal value of (1). The first, $\mathrm{UB}_{\mathrm{LAP}}$, is the value attainable if every image pair were matched independently and optimally, and the second, $\mathrm{UB}_{\mathrm{spec}}$, follows from the spectrum of $\mathbf{W}$. The first bound also bounds the number of improving updates. Each update costs $O(m_i\, d\, \min(m_i, d))$ operations for its assignment problem. The statements and their proofs are given in Appendix A, and the bounds are evaluated on every scene in Appendix B.

4 Experiments

4.1 Benchmark and evaluation protocol

We evaluate on the high-resolution multi-view training sequences of the ETH3D benchmark [5, 6], namely *terrace*, *office*, *electro*, *kicker*, *terrains*, *relief*, *relief_2*, *courtyard*, *delivery_area*, *pipes*, *playground* and *facade*. These scenes span outdoor topography, indoor clutter, repetitive man-made structures and visually ambiguous features, and together they cover the standard multi-view sequences of this benchmark. For every scene we select the $k = 5$ images with the highest mutual co-visibility. The remaining sequence, *meadow*, is degenerate at $k = 5$, because one of its five views retains only 37 keypoints and cannot represent 43 of the 80 universe labels, which removes those correspondences from every path through that view irrespective of the method.

The ground truth is derived from the calibrated multi-view geometry that ETH3D provides, with no correspondence aligned by hand. The keypoints of each image are the 2D observations of reconstructed 3D points, and two keypoints in different images correspond exactly when they observe the same 3D point. The universe consists of the 3D points visible in at least two of the selected views, and a point that is not visible in an image has no row there, so the ground truth is partial and carries no identity bias. The observed matchings are produced by a real feature matcher. A SIFT descriptor is computed at every keypoint, and descriptors are matched across each image pair by a nearest-neighbour test with a ratio test on the distances to the two nearest descriptors. Each image also contains distractor keypoints, drawn from 3D points visible in only one selected view, which have no correct partner anywhere, so $m_i > d$ and the setting is partial. Between 68% and 89% of the observed pairwise matches are correct, depending on the scene, which is the noise regime in which synchronization is meaningful and methods separate. All runs are performed on a machine with an AMD Ryzen 5 5600H CPU and 8 GB of RAM.

4.2 Competing methods and metrics

We compare against eight baselines, namely Spectral [16], NmfSync [36], Stiefel [37], FCC [38], MatchFAME [39], the local-search solvers GM-SEQSEQ and GM-PARPAR [8], and GREEDA [40]. What each of them does is described in Appendix C. To make the comparison fair, each is run at the parameter settings its own paper specifies. The thresholded output of FCC assigns one keypoint to as many as five partners and is therefore not a matching, so we report FCC + exact rounding, its scores put through the same exact block LAP applied to every other baseline. Our method starts from the MatchFAME assignment, and its reported runtime includes that initialization. The guarantees hold from any feasible starting point, and Appendix E reports the evaluation from four further initializations.

We report the F-score, the harmonic mean of precision (the fraction of predicted matches that are correct) and recall (the fraction of ground-truth matches that are recovered), together with precision itself, the cyclic consistency, and the runtime. Cyclic consistency is the agreement between $\mathbf{P}_{i\ell}$ and the composition $\mathbf{P}_{ij}\mathbf{P}_{j\ell}$, measured as the intersection over union of their matches and averaged over all ordered triplets (i, j, ℓ) of distinct images. It is worth mentioning that cycle consistency is a property that every valid solution must satisfy, which is why every method we compare returns a synchronized matching.

4.3 Quantitative results

Table 1 reports the results on *office* and *pipes*, and Tables S2–S4 in Appendix D report every other scene.

Accuracy. As we can see in Table 1, our method attains the best F-score and the best precision on both scenes, 76.0% and 76.4% on *office* and 77.7% and 77.9% on *pipes*, against 68.0% and 68.4%, and 72.6% and 72.8%, for the strongest baseline, NmfSync. The continuous and score-based methods sit between 64.6% and 68.0% in F-score on *office* and between 65.9% and 72.6% on *pipes*. The local-search solvers GM-SEQSEQ, GM-PARPAR and GREEDA reach an F-score of 63.9–65.5% and 66.5–69.3% at a precision of only 56.8–58.2% and 59.0–61.4%, because they assign every keypoint, including the distractors that have no correct partner anywhere. They consequently return more matches than there are labels, and the additional matches are mostly wrong. Our method assigns exactly as many keypoints as the universe can hold, and it gives each one the label with the most support across the collection, which is what lifts precision and F-score together. The same holds across the other scenes, where our method attains the best F-score and the best precision on every sequence except *electro*, on which the universe-based methods all reach 89–93% and ours is on par with the strongest of them (Appendix D).

Consistency and efficiency. Our method is cycle-consistent by construction, and so are the universe-based baselines. The local-search solvers, which carry no universe variable, fall slightly short of full consistency on several scenes. Our runtime, initialization included, is 0.059 s on *office*, 0.060 s on *pipes*, and between 0.043 and 0.060 s on every scene. On both scenes this is four times faster than NmfSync, the strongest baseline in accuracy, and between five and eleven times faster than the local-search solvers. The only faster method, MatchFAME, is precisely the starting point that our

Table 1: Results on the ETH3D sequences *office* and *pipes*, with the best value in bold and the second best underlined.

Dataset	Method	F-score (%) ↑	Precision (%) ↑	Cyclic (%) ↑	Runtime (s) ↓
office	Spectral (2013) [16]	67.8	68.2	**100.0**	0.077
	NmfSync (2019) [36]	<u>68.0</u>	<u>68.4</u>	**100.0**	0.252
	Stiefel (2021) [37]	66.0	66.3	**100.0**	0.261
	FCC + exact rounding (2021) [38]	64.6	65.0	**100.0**	0.165
	MatchFAME (2022) [39]	67.1	67.5	**100.0**	**0.043**
	GM-SEQSEQ (2024) [8]	65.5	58.2	99.9	0.322
	GM-PARPAR (2024) [8]	65.2	57.9	99.6	0.379
	GREEDA (2025) [40]	63.9	56.8	99.9	0.658
	Ours	**76.0**	**76.4**	**100.0**	<u>0.059</u>
pipes	Spectral (2013) [16]	70.3	70.5	**100.0**	0.074
	NmfSync (2019) [36]	<u>72.6</u>	<u>72.8</u>	**100.0**	0.247
	Stiefel (2021) [37]	70.1	70.3	**100.0**	0.306
	FCC + exact rounding (2021) [38]	70.5	70.7	**100.0**	0.165
	MatchFAME (2022) [39]	65.9	66.1	**100.0**	**0.047**
	GM-SEQSEQ (2024) [8]	67.3	59.6	**100.0**	0.334
	GM-PARPAR (2024) [8]	66.5	59.0	99.7	0.393
	GREEDA (2025) [40]	69.3	61.4	**100.0**	0.669
	Ours	**77.7**	**77.9**	**100.0**	<u>0.060</u>

ascent improves by 8.9 points of F-score on *office* and by 11.8 points on *pipes*, at an additional cost of 16 and 13 ms.

4.4 Qualitative comparison

Figures 1 and 2 show the geometric ground truth on one image pair of *office* and of *pipes* alongside the correspondences each method recovers. The pairs are selected by the rule stated in Appendix D, and the figures of all other scenes are given there. Of the 82 ground-truth matches on this pair, our method recovers 64 correct matches against 16 wrong ones, as many correct matches as any method and by far the fewest wrong ones. The continuous and score-based methods recover 49–53 correct matches with 27–31 wrong ones. The local-search solvers find up to 64 correct matches, but at 41–46 wrong ones, since they also match the distractors. Their errors appear as long red diagonals, many of which start at isolated keypoints on the walls and the ceiling that have no partner in the other view. The same pattern holds on *pipes*, where our method makes 19 errors among its 61 correct matches, against 26–42 errors for the continuous and score-based methods, which find at most 54 correct matches, and 42–47 errors for the local-search solvers, which find up to 62 correct matches.

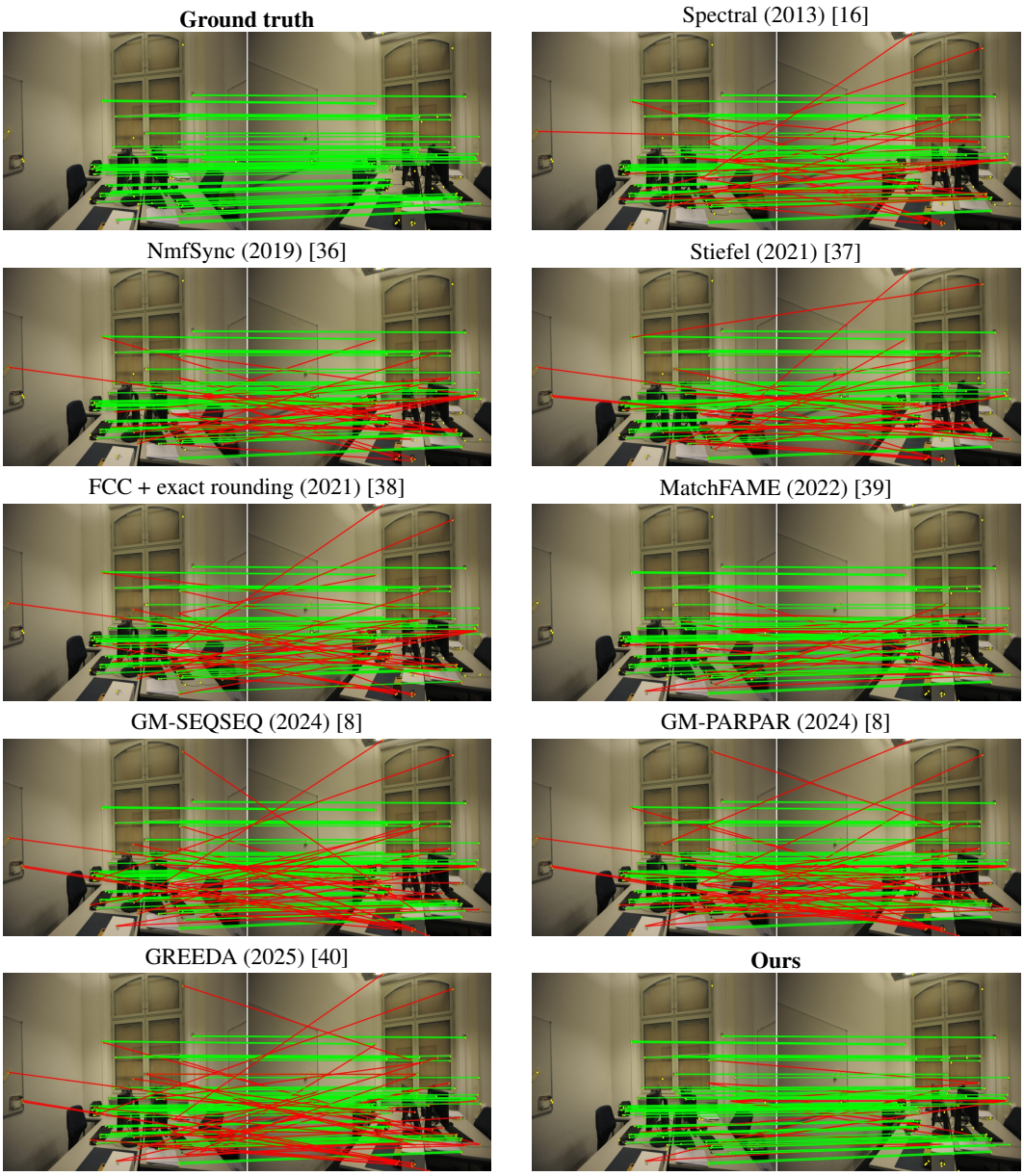

Figure 1: Correspondences recovered on one image pair of *office*. Green lines agree with the geometric ground truth, red lines are wrong, and yellow dots mark the keypoints.

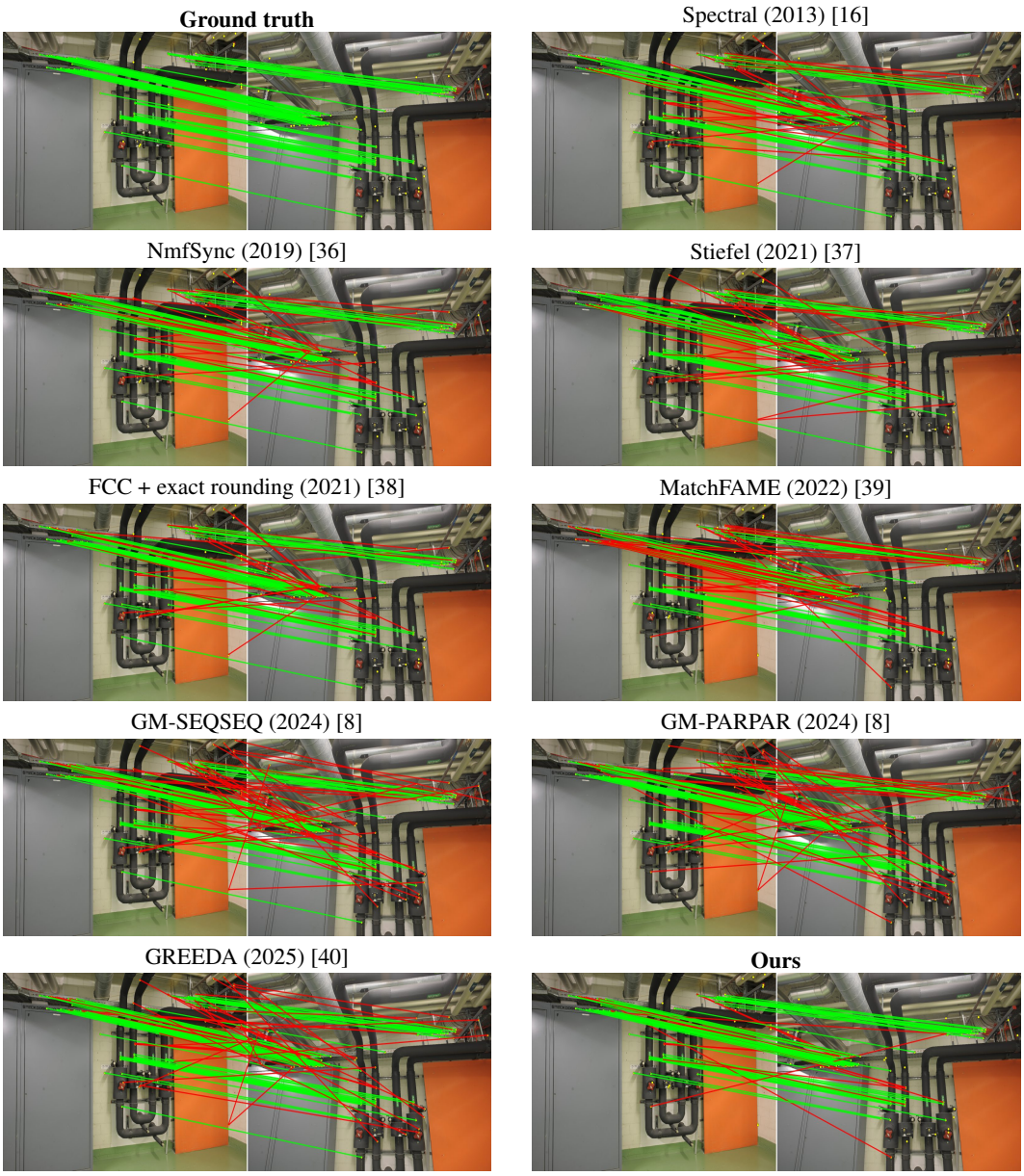

Figure 2: Correspondences recovered on one image pair of *pipes*. Green lines agree with the geometric ground truth, red lines are wrong, and yellow dots mark the keypoints.

5 Limitations

Our problem is non-convex, where no algorithm can guarantee a global maximizer [11]. We establish monotone ascent and finite termination, with each per-image subproblem solved to its exact optimum, and only block-wise optimality of the limit can be proved.

6 Conclusion

We have presented a direct combinatorial method for permutation synchronization that operates entirely in the space of partial permutation matrices and never passes through a relaxation that loses tightness. We showed that feasibility alone forces $\mathbf{P}_i\mathbf{P}_i^T$ to be diagonal, so the per-image restriction of the objective is exactly linear for any symmetric observation matrix. Each update is therefore a rectangular linear assignment problem with a totally unimodular constraint matrix, which is solved exactly and needs no rounding, and every iterate is cycle-consistent by construction. We proved that the objective increases monotonically, that the method terminates after finitely many updates, and that its limit is block-wise optimal when ties are resolved in favour of the current assignment, and we derived computable upper bounds on the optimal objective. On the multi-view sequences of the ETH3D benchmark, with ground truth derived from calibrated 3D geometry, our method outperforms state-of-the-art synchronization and multi-graph matching methods in both F-score and precision.

Acknowledgments and Disclosure of Funding

The authors acknowledge the High Performance Computing (HPC) Center at Sharif University of Technology for providing computational resources that have contributed to this research. This work is supported by INSF, Iran National Science Foundation, grant number 4032041. The authors declare no competing interests.

References

[1] T. F. Cootes and C. J. Taylor, "Active shape models—'smart snakes'," in *BMVC92: Proceedings of the British Machine Vision Conference, organised by the British Machine Vision Association 22–24 September 1992 Leeds*, pp. 266–275, Springer, 1992.

[2] T. Heimann and H.-P. Meinzer, "Statistical shape models for 3d medical image segmentation: a review," *Medical image analysis*, vol. 13, no. 4, pp. 543–563, 2009.

[3] J. Berclaz, F. Fleuret, E. Türetken, and P. Fua, "Multiple object tracking using k-shortest paths optimization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 9, pp. 1806–1819, 2011.

[4] L. Zhang, Y. Li, and R. Nevatia, "Global data association for multi-object tracking using network flows," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–8, 2008.

[5] T. Schöps, J. L. Schönberger, S. Galliani, T. Sattler, K. Schindler, M. Pollefeys, and A. Geiger, "A multi-view stereo benchmark with high-resolution images and multi-camera videos," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[6] T. Schöps, T. Sattler, and M. Pollefeys, "BAD SLAM: Bundle adjusted direct RGB-D SLAM," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[7] J. Yan, X.-C. Yin, W. Lin, C. Deng, H. Zha, and X. Yang, "A short survey of recent advances in graph matching," in *Proceedings of the 2016 ACM on international conference on multimedia retrieval*, pp. 167–174, 2016.

[8] M. Kahl, S. Stricker, L. Hutschenreiter, F. Bernard, and B. Savchynskyy, "Unlocking the potential of operations research for multi-graph matching," 2024.

[9] A. Sotiras, C. Davatzikos, and N. Paragios, "Deformable medical image registration: A survey," *IEEE Transactions on Medical Imaging*, vol. 32, no. 7, pp. 1153–1190, 2013.

[10] B. Zitová and J. Flusser, "Image registration methods: A survey," *Image and Vision Computing*, vol. 21, no. 11, pp. 977–1000, 2003.

[11] S. Sahni and T. Gonzalez, "P-complete approximation problems," *Journal of the ACM (JACM)*, vol. 23, no. 3, pp. 555–565, 1976.

[12] I. Kezurer, S. Z. Kovalsky, R. Basri, and Y. Lipman, "Tight relaxation of quadratic matching," in *Computer graphics forum*, vol. 34, pp. 115–128, Wiley Online Library, 2015.

[13] J. Yan, M. Cho, H. Zha, X. Yang, and S. M. Chu, "Multi-graph matching via affinity optimization with graduated consistency regularization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 6, pp. 1228–1242, 2016.

[14] F. Bernard, C. Theobalt, and M. Moeller, "DS*: Tighter lifting-free convex relaxations for quadratic matching problems," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[15] Q.-X. Huang and L. Guibas, "Consistent shape maps via semidefinite programming," in *Computer graphics forum*, vol. 32, pp. 177–186, Wiley Online Library, 2013.

[16] D. Pachauri, R. Kondor, and V. Singh, "Solving the multi-way matching problem by permutation synchronization," in *Advances in Neural Information Processing Systems* (C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, eds.), vol. 26, Curran Associates, Inc., 2013.

[17] F. Bernard, J. Thunberg, P. Gemmar, F. Hertel, A. Husch, and J. Goncalves, "A solution for multi-alignment by transformation synchronisation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[18] F. Arrigoni, B. Rossi, and A. Fusiello, "Spectral synchronization of multiple views in SE(3)," *SIAM Journal on Imaging Sciences*, vol. 9, no. 4, pp. 1963–1990, 2016.

[19] X. Huang, Z. Liang, X. Zhou, Y. Xie, L. J. Guibas, and Q. Huang, "Learning transformation synchronization," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8082–8091, 2019.

[20] Q. Huang, Z. Liang, H. Wang, S. Zuo, and C. Bajaj, "Tensor maps for synchronizing heterogeneous shape collections," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 4, pp. 1–18, 2019.

[21] R. Huang, J. Ren, P. Wonka, and M. Ovsjanikov, "Consistent ZoomOut: Efficient spectral map synchronization," in *Computer Graphics Forum*, vol. 39, pp. 265–278, Wiley Online Library, 2020.

[22] M. Gao, Z. Lahner, J. Thunberg, D. Cremers, and F. Bernard, "Isometric multi-shape matching," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14183–14193, 2021.

[23] X. Zhou, M. Zhu, and K. Daniilidis, "Multi-image matching via fast alternating minimization," in *Proceedings of the IEEE international conference on computer vision*, pp. 4032–4040, 2015.

[24] R. Tron, X. Zhou, C. Esteves, and K. Daniilidis, "Fast multi-image matching via density-based clustering," in *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 4077–4086, 2017.

[25] F. Bernard, J. Thunberg, P. Swoboda, and C. Theobalt, "HiPPI: Higher-order projected power iterations for scalable multi-matching," in *Proceedings of the ieee/cvf international conference on computer vision*, pp. 10284–10293, 2019.

[26] T. Birdal and U. Simsekli, "Probabilistic permutation synchronization using the Riemannian structure of the Birkhoff polytope," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11097–11108, 2019.

[27] T. Birdal, V. Golyanik, C. Theobalt, and L. J. Guibas, "Quantum permutation synchronization," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 13122–13133, 2021.

[28] P. Swoboda, D. Kainmüller, A. Mokarian, C. Theobalt, and F. Bernard, "A convex relaxation for multi-graph matching," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11156–11165, 2019.

[29] V. Govindu, "Lie-algebraic averaging for globally consistent motion estimation," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, vol. 1, pp. 684–691, 2004.

[30] A. Chatterjee and V. M. Govindu, "Efficient and robust large-scale rotation averaging," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2013.

[31] F. Arrigoni, E. Maset, and A. Fusiello, "Synchronization in the symmetric inverse semigroup," in *Image Analysis and Processing-ICIAP 2017: 19th International Conference, Catania, Italy, September 11-15, 2017, Proceedings, Part II 19*, pp. 70–81, Springer, 2017.

[32] J. Thunberg, F. Bernard, and J. Goncalves, "Distributed methods for synchronization of orthogonal matrices over graphs," *Automatica*, vol. 80, pp. 243–252, 2017.

[33] L. Wang and A. Singer, "Exact and stable recovery of rotations for robust synchronization," *Information and Inference: A Journal of the IMA*, vol. 2, no. 2, pp. 145–193, 2013.

[34] Y. Shen, Q. Huang, N. Srebro, and S. Sanghavi, "Normalized spectral map synchronization," in *Advances in Neural Information Processing Systems* (D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, eds.), vol. 29, Curran Associates, Inc., 2016.

[35] E. Maset, F. Arrigoni, and A. Fusiello, "Practical and efficient multi-view matching," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[36] F. Bernard, J. Thunberg, J. Goncalves, and C. Theobalt, "Synchronisation of partial multi-matchings via non-negative factorisations," *Pattern Recognition*, vol. 92, pp. 146–155, 2019.

[37] F. Bernard, D. Cremers, and J. Thunberg, "Sparse quadratic optimisation over the Stiefel manifold with application to permutation synchronisation," *Advances in Neural Information Processing Systems*, vol. 34, pp. 25256–25266, 2021.

[38] Y. Shi, S. Li, T. Maunu, and G. Lerman, "Scalable cluster-consistency statistics for robust multi-object matching," in *2021 International Conference on 3D Vision (3DV)*, pp. 352–360, 2021.

[39] S. Li, Y. Shi, and G. Lerman, "Fast, accurate and memory-efficient partial permutation synchronization," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15714–15722, 2022.

[40] M. Kahl, S. Stricker, L. Hutschenreiter, F. Bernard, C. Rother, and B. Savchynskyy, "Towards optimizing large-scale multi-graph matching in bioimaging," in *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11569–11578, 2025.

[41] A. Schrijver, *Theory of linear and integer programming*. USA: John Wiley & Sons, Inc., 1986.

[42] G. L. Nemhauser and L. A. Wolsey, *Integer and combinatorial optimization*. John Wiley & Sons, 1999.

[43] R. Jonker and A. Volgenant, "A shortest augmenting path algorithm for dense and sparse linear assignment problems," *Computing*, vol. 38, no. 4, pp. 325–340, 1987.

[44] Y. Sun, P. Babu, and D. P. Palomar, "Majorization-minimization algorithms in signal processing, communications, and machine learning," *IEEE Transactions on Signal Processing*, vol. 65, no. 3, pp. 794–816, 2017.

[45] S. Pandey, L. Magri, F. Arrigoni, and V. Golyanik, "Outlier-robust multi-model fitting on quantum annealers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 2846–2855, June 2025.

[46] C. Holtz, P. Chen, Z. Wan, C.-K. Cheng, and G. Mishne, "Continuous partitioning for graph-based semi-supervised learning," in *Advances in Neural Information Processing Systems* (A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, eds.), vol. 37, pp. 5615–5645, Curran Associates, Inc., 2024.

[47] M. Chen, J. Chun, S. Xiang, L. Wei, Y. Du, Q. Wan, Y. Chen, and Y. Chen, "Learning to solve quadratic unconstrained binary optimization in a classification way," in *Advances in Neural Information Processing Systems* (A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, eds.), vol. 37, pp. 114478–114509, Curran Associates, Inc., 2024.

Technical Appendices

A Proofs of the Theoretical Results

Throughout this appendix, $\mathbf{P} \in \mathbb{U}$ is partitioned into blocks $\mathbf{P}_1, \dots, \mathbf{P}_k$ with $\mathbf{P}_i \in \{0,1\}^{m_i \times d}$, and $\mathbf{W}$ into blocks $\mathbf{W}_{ab} \in \mathbb{R}^{m_a \times m_b}$ with $\mathbf{W}_{ba} = \mathbf{W}_{ab}^T$. Entries are written $(\mathbf{P}_i)_{rc}$.

A.1 Proof of Lemma 3.1

Proof. Let $\mathbf{P}_i \in \mathbb{P}_{m_i d}$ and $\mathbf{a} := \mathbf{P}_i \mathbb{1}_d$, so that $a_r = \sum_c (\mathbf{P}_i)_{rc} \in \{0,1\}$. For every pair of rows r, s,

$$(\mathbf{P}_i \mathbf{P}_i^T)_{rs} = \sum_{c=1}^{d} (\mathbf{P}_i)_{rc} (\mathbf{P}_i)_{sc}. \tag{S1}$$

If $r \neq s$, the column constraint $\mathbf{P}_i^T \mathbb{1}_{m_i} \leq \mathbb{1}_d$ gives $(\mathbf{P}_i)_{rc} + (\mathbf{P}_i)_{sc} \leq 1$ for every c, and since both entries are binary, $(\mathbf{P}_i)_{rc}(\mathbf{P}_i)_{sc} = 0$. Every summand of (S1) is zero, hence $(\mathbf{P}_i \mathbf{P}_i^T)_{rs} = 0$. If $r = s$, binarity gives $(\mathbf{P}_i)_{rc}^2 = (\mathbf{P}_i)_{rc}$, hence $(\mathbf{P}_i \mathbf{P}_i^T)_{rr} = \sum_c (\mathbf{P}_i)_{rc} = a_r$. Therefore $\mathbf{P}_i \mathbf{P}_i^T = \mathrm{Diag}(\mathbf{a})$. By the cyclic property of the trace,

$$\begin{aligned} \mathrm{tr}\big(\mathbf{P}_i^T \mathbf{W}_{ii} \mathbf{P}_i\big) &= \mathrm{tr}\big(\mathbf{W}_{ii} \mathbf{P}_i \mathbf{P}_i^T\big) = \mathrm{tr}\big(\mathbf{W}_{ii}\, \mathrm{Diag}(\mathbf{a})\big) = \sum_{r=1}^{m_i} (\mathbf{W}_{ii})_{rr}\, a_r \\ &= \sum_{r=1}^{m_i} \sum_{c=1}^{d} (\mathbf{W}_{ii})_{rr} (\mathbf{P}_i)_{rc} = \big\langle \mathrm{diag}(\mathbf{W}_{ii}) \mathbb{1}_d^T,\, \mathbf{P}_i \big\rangle. \end{aligned}$$

■

A.2 Proof of Proposition 1

Proof. Expanding the quadratic term block by block,

$$\mathrm{tr}\big(\mathbf{P}^T \mathbf{W} \mathbf{P}\big) = \sum_{a=1}^{k} \sum_{b=1}^{k} \mathrm{tr}\big(\mathbf{P}_a^T \mathbf{W}_{ab} \mathbf{P}_b\big).$$

The summands that contain $\mathbf{P}_i$ are the one with $a = b = i$ and, for every $j \neq i$, the two with $(a,b) = (i,j)$ and $(a,b) = (j,i)$. Since $\mathbf{W}_{ji} = \mathbf{W}_{ij}^T$,

$$\mathrm{tr}\big(\mathbf{P}_j^T \mathbf{W}_{ji} \mathbf{P}_i\big) = \mathrm{tr}\big((\mathbf{P}_i^T \mathbf{W}_{ij} \mathbf{P}_j)^T\big) = \mathrm{tr}\big(\mathbf{P}_i^T \mathbf{W}_{ij} \mathbf{P}_j\big),$$

and therefore

$$\mathrm{tr}\big(\mathbf{P}^T \mathbf{W} \mathbf{P}\big) = \mathrm{tr}\big(\mathbf{P}_i^T \mathbf{W}_{ii} \mathbf{P}_i\big) + 2 \sum_{j \neq i} \mathrm{tr}\big(\mathbf{P}_i^T \mathbf{W}_{ij} \mathbf{P}_j\big) + \sum_{a \neq i} \sum_{b \neq i} \mathrm{tr}\big(\mathbf{P}_a^T \mathbf{W}_{ab} \mathbf{P}_b\big).$$

The middle term equals $2\,\mathrm{tr}(\mathbf{P}_i^T \mathbf{E}_i) = \langle 2\mathbf{E}_i, \mathbf{P}_i \rangle$, and by Lemma 3.1 the first equals $\langle \mathrm{diag}(\mathbf{W}_{ii}) \mathbb{1}_d^T, \mathbf{P}_i \rangle$. The linear term splits as $\langle \mathbf{C}, \mathbf{P} \rangle = \langle \mathbf{C}_i, \mathbf{P}_i \rangle + \sum_{j \neq i} \langle \mathbf{C}_j, \mathbf{P}_j \rangle$. Collecting terms gives (3) with

$$\kappa_i = \sum_{a \neq i} \sum_{b \neq i} \mathrm{tr}\big(\mathbf{P}_a^T \mathbf{W}_{ab} \mathbf{P}_b\big) + \sum_{j \neq i} \langle \mathbf{C}_j, \mathbf{P}_j \rangle,$$

which does not depend on $\mathbf{P}_i$. ■

A.3 Proof of Theorem 3.2

We first record that appending unit rows preserves total unimodularity.

Lemma A.1. *If $\mathbf{D} \in \mathbb{Z}^{p \times n}$ is totally unimodular, so is $\widetilde{\mathbf{D}} = \begin{bmatrix} \mathbf{D} \\ -\mathbf{I}_n \end{bmatrix}$.*

Proof. Let $\mathbf{B}$ be a square submatrix of $\widetilde{\mathbf{D}}$ of order q. We argue by induction on the number s of rows of $\mathbf{B}$ taken from $-\mathbf{I}_n$. If $s = 0$, $\mathbf{B}$ is a submatrix of $\mathbf{D}$ and $\det \mathbf{B} \in \{-1, 0, 1\}$. If $s \geq 1$, let $\mathbf{r}$ be a row of $\mathbf{B}$ taken from $-\mathbf{I}_n$. It has at most one nonzero entry, equal to -1. If it has none, $\det \mathbf{B} = 0$. Otherwise the Laplace expansion along $\mathbf{r}$ gives $\det \mathbf{B} = \pm \det \mathbf{B}'$, where $\mathbf{B}'$ is a square submatrix of $\widetilde{\mathbf{D}}$ of order $q - 1$ with $s - 1$ rows from $-\mathbf{I}_n$, and $\det \mathbf{B}' \in \{-1, 0, 1\}$ by the induction hypothesis. ■

With Lemma A.1, integrality of the vertices follows from Cramer's rule.

Proof. Write the polyhedron as $\{\mathbf{x} \in \mathbb{R}^n : \widetilde{\mathbf{D}}\mathbf{x} \leq \widetilde{\mathbf{b}}\}$ with $\widetilde{\mathbf{D}}$ as in Lemma A.1 and $\widetilde{\mathbf{b}} = [\mathbf{b}^T, \mathbf{0}^T]^T \in \mathbb{Z}^{p+n}$. Let $\bar{\mathbf{x}}$ be a vertex. By the definition of a vertex, there are n linearly independent rows of $\widetilde{\mathbf{D}}$ at which the constraint is tight, that is, a nonsingular square submatrix $\mathbf{B}$ of order n and the corresponding subvector $\widetilde{\mathbf{b}}_B$ with $\mathbf{B}\bar{\mathbf{x}} = \widetilde{\mathbf{b}}_B$. By Lemma A.1, $\det \mathbf{B} \in \{-1, 1\}$. By Cramer's rule,

$$\bar{x}_j = \frac{\det \mathbf{B}_j}{\det \mathbf{B}}, \qquad j = 1, \ldots, n,$$

where $\mathbf{B}_j$ is obtained from $\mathbf{B}$ by replacing its j-th column by $\widetilde{\mathbf{b}}_B$. Since $\mathbf{B}_j$ is an integer matrix, $\det \mathbf{B}_j \in \mathbb{Z}$, and hence $\bar{x}_j \in \mathbb{Z}$ for every j. ■

A.4 Proof of Lemma 3.3

Proof. Index the columns of $\mathbf{D}$ by pairs (r, c), with $r \in \{1, \ldots, m\}$ a row of $\mathbf{P}$ and $c \in \{1, \ldots, d\}$ a label, so that column (r, c) acts on $x_{(c-1)m+r} = P_{rc}$, and let $\iota(r)$ denote the image that row r belongs to. With $\mathbf{e}_\cdot$ the canonical basis vectors of the appropriate dimension, column (r, c) of the first block of (4) is $(\mathbb{1}_d^T \otimes \mathbf{I}_m)(\mathbf{e}_c \otimes \mathbf{e}_r) = \mathbf{e}_r$, and of the second block $(\mathbf{I}_d \otimes \mathbf{J})(\mathbf{e}_c \otimes \mathbf{e}_r) = \mathbf{e}_c \otimes \mathbf{J}\mathbf{e}_r = \mathbf{e}_c \otimes \mathbf{e}_{\iota(r)}$. Hence every column of $\mathbf{D}$ has exactly two nonzero entries, both equal to 1, one in the first block of m rows and one in the second block of kd rows.

Let $\mathbf{B}$ be a square submatrix of $\mathbf{D}$ of order q. We prove $\det \mathbf{B} \in \{-1, 0, 1\}$ by induction on q. For $q = 1$ the entry of $\mathbf{B}$ is 0 or 1. For $q \geq 2$, every column of $\mathbf{B}$ has at most two nonzero entries, and three cases arise.

(a) Some column of $\mathbf{B}$ is zero. Then $\det \mathbf{B} = 0$.

(b) Some column of $\mathbf{B}$ has exactly one nonzero entry, equal to 1. The Laplace expansion along that column gives $\det \mathbf{B} = \pm \det \mathbf{B}'$ for a square submatrix $\mathbf{B}'$ of $\mathbf{D}$ of order $q - 1$, and $\det \mathbf{B}' \in \{-1, 0, 1\}$ by the induction hypothesis.

(c) Every column of $\mathbf{B}$ has exactly two nonzero entries. Each column of $\mathbf{D}$ has one unit entry in each block, so each column of $\mathbf{B}$ has one unit entry among the rows of $\mathbf{B}$ drawn from the first block and one among those drawn from the second. Let $\mathbf{u}$ be the vector with entries $+1$ on the rows of $\mathbf{B}$ drawn from the first block and -1 on those drawn from the second. Then $\mathbf{u}^T\mathbf{B} = \mathbf{0}^T$ with $\mathbf{u} \neq \mathbf{0}$, so the rows of $\mathbf{B}$ are linearly dependent and $\det \mathbf{B} = 0$.

In every case $\det \mathbf{B} \in \{-1, 0, 1\}$, which proves that $\mathbf{D}$ is totally unimodular. ■

A.5 Proof of Proposition 2

Proof. Let $Q := \{\mathbf{x} \geq \mathbf{0} : \mathbf{D}\mathbf{x} \leq \mathbf{b}\}$. By Lemma 3.3 and Theorem 3.2, every vertex of Q is integral. Every $\mathbf{x} \in Q$ satisfies $0 \leq x_j \leq 1$, since each coordinate appears with coefficient 1 in a row constraint of right-hand side 1, so Q is a polytope and its vertices are binary.

For every $\mathbf{x} = \mathrm{vec}(\mathbf{P}) \in Q$ and every image i, define $g_i(\mathbf{x}) := \mathbb{1}_{m_i}^T \mathbf{P}_i \mathbb{1}_d$. Summing the row constraints of the rows of image i gives $g_i(\mathbf{x}) \leq m_i$, and summing the capacity constraints of image i over the d labels gives $g_i(\mathbf{x}) \leq d$. Hence $g_i(\mathbf{x}) \leq \min(m_i, d)$ is a valid inequality for Q, and $F_i := \{\mathbf{x} \in Q : g_i(\mathbf{x}) = \min(m_i, d)\}$ is a face of Q. The feasible set of (5) is $F := F_1 \cap \cdots \cap F_k$, an intersection of faces, and hence itself a face of Q. It is nonempty, since the matrix $\mathbf{P}$ whose block i matches its first $\min(m_i, d)$ keypoints to the first $\min(m_i, d)$ labels satisfies all constraints with equality in each g_i. The vertices of a face of a polytope are vertices of the polytope, so every vertex of F is binary. A linear function attains its maximum over the nonempty polytope F at a vertex $\mathbf{x}^\star = \mathrm{vec}(\mathbf{P}^\star)$. The matrix $\mathbf{P}^\star$ is binary, each of its rows has at most one unit entry, each label is used at most once in each block, and block i has exactly $\min(m_i, d)$ unit entries. Therefore $\mathbf{P}_i^\star \in \overline{\mathbb{P}}_{m_i d}$ for every i, that is, $\mathbf{P}^\star \in \mathbb{U}$.

For the decoupling, every row constraint and every capacity constraint involves the entries of a single block, and so does every equality $g_i(\mathbf{x}) = \min(m_i, d)$. The objective separates as $\langle \mathbf{C}, \mathbf{P} \rangle = \sum_{i=1}^{k} \langle \mathbf{C}_i, \mathbf{P}_i \rangle$. Problem (5) is therefore the sum of k independent linear programs, the i-th of which maximizes $\langle \mathbf{C}_i, \mathbf{P}_i \rangle$ over the i-th face. By the argument above applied with $k = 1$, each attains its maximum at an element of $\overline{\mathbb{P}}_{m_i d}$, which selects $\min(m_i, d)$ entries of $\mathbf{C}_i$ with at most one per row and per column, that is, a rectangular linear assignment problem. ■

A.6 The block update as a minorization–maximization step

Minorization–maximization is a principle for maximizing an objective F over a set $\mathcal{X}$ [44]. At the current iterate $\mathbf{X}^{(t)}$ one builds a surrogate $g(\cdot \mid \mathbf{X}^{(t)})$ that satisfies

$$g(\mathbf{X} \mid \mathbf{X}^{(t)}) \leq F(\mathbf{X}) \quad \text{for all } \mathbf{X} \in \mathcal{X}, \qquad g(\mathbf{X}^{(t)} \mid \mathbf{X}^{(t)}) = F(\mathbf{X}^{(t)}), \tag{S2}$$

and takes $\mathbf{X}^{(t+1)} \in \arg\max_{\mathbf{X} \in \mathcal{X}} g(\mathbf{X} \mid \mathbf{X}^{(t)})$. The two conditions give $F(\mathbf{X}^{(t+1)}) \geq g(\mathbf{X}^{(t+1)} \mid \mathbf{X}^{(t)}) \geq g(\mathbf{X}^{(t)} \mid \mathbf{X}^{(t)}) = F(\mathbf{X}^{(t)})$, so every step ascends. The following proposition shows that each update of Algorithm 1 is such a step, taken on one block, with a surrogate that coincides with the objective.

Proposition 3 (The block update is a tight minorization–maximization step)**.** *Fix every block of $\mathbf{P}^{(t)}$ except block i, let $\mathcal{X}_i = \{\mathbf{P} \in \mathbb{U} : \mathbf{P}_j = \mathbf{P}_j^{(t)} \text{ for all } j \neq i\}$, and define*

$$g_i(\mathbf{P} \mid \mathbf{P}^{(t)}) := \left\langle \mathbf{V}_i^{(t)}, \mathbf{P}_i \right\rangle + \kappa_i^{(t)}, \tag{S3}$$

with $\mathbf{V}_i^{(t)}$ and $\kappa_i^{(t)}$ given by Proposition 1 at $\mathbf{P}^{(t)}$. Then g_i satisfies both conditions (S2) *on $\mathcal{X}_i$ with F of* (2)*, and in fact $g_i(\mathbf{P} \mid \mathbf{P}^{(t)}) = F(\mathbf{P})$ for every $\mathbf{P} \in \mathcal{X}_i$. Its maximizer over $\mathcal{X}_i$ is the update* (6) *when $\mathbf{C} = \mathbf{0}$ and $\mathbf{W}_{ii} = \mathbf{I}_{m_i}$.*

Proof. Every $\mathbf{P} \in \mathcal{X}_i$ agrees with $\mathbf{P}^{(t)}$ outside block i, so $\mathbf{E}_i$, $\mathbf{V}_i$ and κ_i of Proposition 1 evaluated at $\mathbf{P}$ equal $\mathbf{E}_i^{(t)}$, $\mathbf{V}_i^{(t)}$ and $\kappa_i^{(t)}$. Proposition 1 then gives $F(\mathbf{P}) = \langle \mathbf{V}_i^{(t)}, \mathbf{P}_i \rangle + \kappa_i^{(t)} = g_i(\mathbf{P} \mid \mathbf{P}^{(t)})$ for every $\mathbf{P} \in \mathcal{X}_i$. Equality everywhere implies the first condition of (S2), and taking $\mathbf{P} = \mathbf{P}^{(t)} \in \mathcal{X}_i$ gives the second. When $\mathbf{C} = \mathbf{0}$ and $\mathbf{W}_{ii} = \mathbf{I}_{m_i}$, $\mathbf{V}_i^{(t)} = 2\mathbf{E}_i^{(t)} + \mathbb{1}_{m_i} \mathbb{1}_d^T$, and on $\overline{\mathbb{P}}_{m_i d}$ the second term contributes the constant $\min(m_i, d)$, so maximizing g_i over $\mathcal{X}_i$ is maximizing $\langle \mathbf{E}_i^{(t)}, \mathbf{P}_i \rangle$ over $\overline{\mathbb{P}}_{m_i d}$, which is (6). ■

Algorithm 1 therefore applies the minorization–maximization principle to one block at a time, cycling over the blocks. Because Lemma 3.1 makes the restriction of the objective to one block exactly linear, the surrogate (S3) coincides with the objective on all of $\mathcal{X}_i$, so the minorization is tight and each step maximizes the objective itself over the block.

A.7 Monotone ascent, finite termination and cost

We write $\mathbf{P}^{(t)}$ for the iterate after the t-th block update of Algorithm 1, and $f(\mathbf{P}) = \operatorname{tr}(\mathbf{P}^T \mathbf{W} \mathbf{P})$.

Proposition 4 (Monotone ascent and finite termination)**.** *Let $\mathbf{W}_{ii} = \mathbf{I}_{m_i}$ for every i. Then*

- *(i) every block update satisfies $f(\mathbf{P}^{(t+1)}) \geq f(\mathbf{P}^{(t)})$;*
- *(ii) if $\mathbf{W} \in \{0, 1\}^{m \times m}$, the number of strictly improving updates is at most $\mathrm{UB}_{\mathrm{LAP}} - f(\mathbf{P}^{(0)})$, with $\mathrm{UB}_{\mathrm{LAP}}$ as in Proposition 7, so Algorithm 1 terminates after at most $\mathrm{UB}_{\mathrm{LAP}} - f(\mathbf{P}^{(0)}) + 1$ passes;*
- *(iii) if each update retains the incumbent block whenever the incumbent attains the maximum of* (6)*, the iterates become constant after finitely many updates, and the limit $\mathbf{P}^\star$ is block-wise optimal, that is, $f(\mathbf{P}^\star) \geq f(\mathbf{P})$ for every $\mathbf{P} \in \mathbb{U}$ that differs from $\mathbf{P}^\star$ in a single block.*

Proof. (i) Let block i be updated at $\mathbf{P} = \mathbf{P}^{(t)}$. By Proposition 1 with $\mathbf{C} = \mathbf{0}$ and $\mathbf{W}_{ii} = \mathbf{I}_{m_i}$, for every $\mathbf{P}'$ that agrees with $\mathbf{P}$ outside block i and has $\mathbf{P}'_i \in \overline{\mathbb{P}}_{m_i d}$,

$$f(\mathbf{P}') = 2\langle \mathbf{E}_i, \mathbf{P}'_i \rangle + \langle \mathbb{1}_{m_i} \mathbb{1}_d^T, \mathbf{P}'_i \rangle + \kappa_i = 2\langle \mathbf{E}_i, \mathbf{P}'_i \rangle + \min(m_i, d) + \kappa_i, \tag{S4}$$

since every element of $\overline{\mathbb{P}}_{m_i d}$ has exactly $\min(m_i, d)$ unit entries. The update returns $\mathbf{P}_i^{\text{new}} \in \arg\max_{\mathbf{X} \in \overline{\mathbb{P}}_{m_i d}} \langle \mathbf{E}_i, \mathbf{X} \rangle$, and the incumbent $\mathbf{P}_i$ is feasible for this maximization, so $\langle \mathbf{E}_i, \mathbf{P}_i^{\text{new}} \rangle \geq \langle \mathbf{E}_i, \mathbf{P}_i \rangle$, and (S4) gives $f(\mathbf{P}^{(t+1)}) \geq f(\mathbf{P}^{(t)})$.

(ii) If $\mathbf{W}$ and $\mathbf{P}$ are integral, $f(\mathbf{P}) = \sum_{a,b} \sum_{r,s,c} (\mathbf{P}_a)_{rc} (\mathbf{W}_{ab})_{rs} (\mathbf{P}_b)_{sc} \in \mathbb{Z}$. Every iterate is integral by Proposition 2, so a strictly improving update raises f by at least 1. By (i) the sequence $f(\mathbf{P}^{(t)})$ is non-decreasing, and by Proposition 7 it is bounded above by $\mathrm{UB}_{\mathrm{LAP}}$. Hence at most $\mathrm{UB}_{\mathrm{LAP}} - f(\mathbf{P}^{(0)})$ updates are strictly improving. Algorithm 1 continues only after a pass that raises f, which requires at least one strictly improving update, so it performs at most $\mathrm{UB}_{\mathrm{LAP}} - f(\mathbf{P}^{(0)})$ passes that raise f, followed by one that does not.

(iii) Under the retention rule, an update changes $\mathbf{P}$ only if it strictly increases f. By (ii) there are finitely many such updates, so after finitely many updates a full pass changes no block, and the iterates are constant from then on. At the limit $\mathbf{P}^\star$, the incumbent $\mathbf{P}_i^\star$ attains $\max_{\mathbf{X} \in \overline{\mathbb{P}}_{m_i d}} \langle \mathbf{E}_i^\star, \mathbf{X} \rangle$ for every i, where $\mathbf{E}_i^\star$ is evaluated at $\mathbf{P}^\star$. For any $\mathbf{P}$ that differs from $\mathbf{P}^\star$ only in block i, (S4) gives $f(\mathbf{P}) - f(\mathbf{P}^\star) = 2\langle \mathbf{E}_i^\star, \mathbf{P}_i - \mathbf{P}_i^\star \rangle \leq 0$. ■

Corollary A.2 (Convergence of the objective)**.** *If $\mathbf{W} \in \{0,1\}^{m \times m}$, the objective values generated by Algorithm 1 are non-decreasing and attain their limit after finitely many updates.*

Proof. The sequence $f(\mathbf{P}^{(t)})$ is non-decreasing by Proposition 4(i), integer-valued, and bounded above by $\mathrm{UB}_{\mathrm{LAP}}$. A non-decreasing sequence of integers that is bounded above takes finitely many values and is therefore eventually constant. ■

Equal-objective updates. A block subproblem may admit several maximizers of equal value. Our implementation accepts an equal-value maximizer that differs from the incumbent, which keeps parts (i) and (ii) of Proposition 4 intact. Such a move changes the evidence $\mathbf{E}_j$ seen by every other block, so it lets the ascent traverse a plateau of equal objective and reach strictly improving updates that retaining the incumbent would not expose.

Proposition 5 (Cost)**.** *One update of block i costs $O(\mathrm{nnz}(\mathbf{W}_{i,\cdot}))$ operations to form $\mathbf{E}_i$, with $\mathrm{nnz}(\mathbf{W}_{i,\cdot})$ the number of nonzeros in the rows of $\mathbf{W}$ that belong to image i, and $O(m_i\, d\, \min(m_i, d))$ operations to solve the rectangular LAP by the shortest-augmenting-path method [43]. If $\mathbf{W} \in \{0,1\}^{m \times m}$, Algorithm 1 performs at most $\mathrm{UB}_{\mathrm{LAP}} - f(\mathbf{P}^{(0)}) + 1$ passes, each costing $\sum_{i=1}^{k} O\big(\mathrm{nnz}(\mathbf{W}_{i,\cdot}) + m_i\, d\, \min(m_i, d)\big)$.*

Proof. The entry $\mathbf{E}_i[r, c] = \sum_{j \neq i} \sum_s (\mathbf{W}_{ij})_{rs} (\mathbf{P}_j)_{sc}$. Each row s of $\mathbf{P}_j$ has at most one unit entry, so every nonzero $(\mathbf{W}_{ij})_{rs}$ contributes to at most one entry of $\mathbf{E}_i$, and forming $\mathbf{E}_i$ requires one addition per nonzero of $\mathbf{W}$ in the rows of image i. The rectangular LAP on the $m_i \times d$ cost matrix is solved by $\min(m_i, d)$ augmentations, each of which is a shortest-path computation on the complete bipartite graph with $m_i + d$ vertices and $m_i d$ edges and costs $O(m_i d)$ [43], which gives $O(m_i\, d\, \min(m_i, d))$ in total. The bound on the number of passes is Proposition 4(ii). ■

Proposition 6 (Stationarity of the terminal point)**.** *Let $\mathbf{P}^\star$ be the limit of Proposition 4(iii). Then for every i and every $\mathbf{P}_i \in \overline{\mathbb{P}}_{m_i d}$,*

$$\big\langle \nabla_{\mathbf{P}_i} f(\mathbf{P}^\star),\ \mathbf{P}_i - \mathbf{P}_i^\star \big\rangle \leq 0,$$

that is, $\mathbf{P}^\star$ satisfies the first-order optimality condition block-wise, in addition to the zeroth-order condition $f(\mathbf{P}) \leq f(\mathbf{P}^\star)$ for every single-block deviation $\mathbf{P}$.

Proof. Since $\mathbf{W}$ is symmetric and $\mathbf{W}_{ii} = \mathbf{I}_{m_i}$, $\nabla_{\mathbf{P}_i} f(\mathbf{P}) = 2(\mathbf{W}_{ii}\mathbf{P}_i + \mathbf{E}_i) = 2(\mathbf{P}_i + \mathbf{E}_i)$. At $\mathbf{P}^\star$,

$$\big\langle \nabla_{\mathbf{P}_i} f(\mathbf{P}^\star), \mathbf{P}_i - \mathbf{P}_i^\star \big\rangle = 2\big\langle \mathbf{P}_i^\star, \mathbf{P}_i \big\rangle - 2\big\langle \mathbf{P}_i^\star, \mathbf{P}_i^\star \big\rangle + 2\big\langle \mathbf{E}_i^\star, \mathbf{P}_i - \mathbf{P}_i^\star \big\rangle.$$

Both $\mathbf{P}_i$ and $\mathbf{P}_i^\star$ are binary with exactly $\min(m_i, d)$ unit entries, so $\langle \mathbf{P}_i^\star, \mathbf{P}_i^\star \rangle = \min(m_i, d)$ and $\langle \mathbf{P}_i^\star, \mathbf{P}_i \rangle \leq \min(m_i, d)$, and the first difference is non-positive. By Proposition 4(iii), $\mathbf{P}_i^\star$ maximizes $\langle \mathbf{E}_i^\star, \cdot \rangle$ over $\overline{\mathbb{P}}_{m_i d}$, so the last term is non-positive. The zeroth-order condition is Proposition 4(iii). ■

A.8 Upper bounds on the optimal objective

Proposition 7 (Upper bounds)**.** *Let L_{ij} be the optimal value of the LAP $\max_{\mathbf{\Pi} \in \mathbb{P}_{m_i m_j}} \langle \mathbf{W}_{ij}, \mathbf{\Pi} \rangle$, and let σ_i be the sum of the $\min(m_i, d)$ largest entries of $\mathrm{diag}(\mathbf{W}_{ii})$, so that $\sigma_i = \min(m_i, d)$ when*

$\mathbf{W}_{ii} = \mathbf{I}_{m_i}$. *Then for every* $\mathbf{P} \in \mathbb{U}$

$$f(\mathbf{P}) \le \sum_{i=1}^{k} \sigma_i + 2 \sum_{1 \le i < j \le k} L_{ij} =: \mathrm{UB}_{\mathrm{LAP}}, \tag{S5}$$

$$f(\mathbf{P}) \le k \sum_{c=1}^{d} \big[\lambda_c(\mathbf{W})\big]_+ =: \mathrm{UB}_{\mathrm{spec}}. \tag{S6}$$

Proof. *Bound* (S5). By the expansion in the proof of Proposition 1,

$$f(\mathbf{P}) = \sum_{i=1}^{k} \mathrm{tr}\big(\mathbf{P}_i^T \mathbf{W}_{ii} \mathbf{P}_i\big) + 2 \sum_{i<j} \mathrm{tr}\big(\mathbf{P}_i^T \mathbf{W}_{ij} \mathbf{P}_j\big) = \sum_{i=1}^{k} \big\langle \mathbf{W}_{ii}, \mathbf{P}_i \mathbf{P}_i^T \big\rangle + 2 \sum_{i<j} \big\langle \mathbf{W}_{ij}, \mathbf{P}_i \mathbf{P}_j^T \big\rangle.$$

For $i \neq j$, the matrix $\mathbf{\Pi}_{ij} := \mathbf{P}_i \mathbf{P}_j^T \in \{0,1\}^{m_i \times m_j}$ has row sums $\mathbf{\Pi}_{ij} \mathbb{1}_{m_j} = \mathbf{P}_i (\mathbf{P}_j^T \mathbb{1}_{m_j}) \le \mathbf{P}_i \mathbb{1}_d \le \mathbb{1}_{m_i}$, and by symmetry column sums at most one, so $\mathbf{\Pi}_{ij} \in \mathbb{P}_{m_i m_j}$ and $\langle \mathbf{W}_{ij}, \mathbf{\Pi}_{ij} \rangle \le L_{ij}$. For the diagonal blocks, Lemma 3.1 gives $\langle \mathbf{W}_{ii}, \mathbf{P}_i \mathbf{P}_i^T \rangle = \langle \mathrm{diag}(\mathbf{W}_{ii}), \mathbf{a}_i \rangle$ with $\mathbf{a}_i = \mathbf{P}_i \mathbb{1}_d \in \{0,1\}^{m_i}$ and, by maximality, $\mathbb{1}^T \mathbf{a}_i = \min(m_i, d)$. The largest value of $\langle \mathrm{diag}(\mathbf{W}_{ii}), \mathbf{a} \rangle$ over binary $\mathbf{a}$ with $\min(m_i, d)$ ones is σ_i. Summing the two bounds proves (S5).

Bound (S6). Every label is used at most once per image, so $\mathbf{S} := \mathbf{P}^T \mathbf{P} = \mathrm{Diag}(n_1, \ldots, n_d)$ with $n_c \le k$. Let $\mathcal{C} = \{c : n_c > 0\}$ and let $\mathbf{u}_c = \mathbf{P}\mathbf{e}_c / \sqrt{n_c}$ for $c \in \mathcal{C}$, which are orthonormal. Then

$$f(\mathbf{P}) = \sum_{c \in \mathcal{C}} n_c \, \mathbf{u}_c^T \mathbf{W} \mathbf{u}_c \le k \sum_{c \in \mathcal{C}} \big[\mathbf{u}_c^T \mathbf{W} \mathbf{u}_c\big]_+ \le k \sum_{c=1}^{d} \big[\lambda_c(\mathbf{W})\big]_+,$$

where the first inequality uses $0 < n_c \le k$, and the second follows from Ky Fan's maximum principle applied to the at most d orthonormal vectors $\mathbf{u}_c$ with $\mathbf{u}_c^T \mathbf{W} \mathbf{u}_c > 0$. ■

The bound $\mathrm{UB}_{\mathrm{LAP}}$ has a transparent interpretation. It is the value attainable if cycle consistency were dropped and every image pair were matched independently and optimally, so the gap between a cycle-consistent solution and $\mathrm{UB}_{\mathrm{LAP}}$ measures the price of consistency on a noisy instance. Its diagonal contribution is exact, since by Lemma 3.1 every feasible $\mathbf{P}$ contributes precisely $\sum_i \sigma_i$ there, and all remaining slack sits in the off-diagonal terms.

B Evaluation of the Upper Bounds

Table S1 reports, for every scene, the two bounds of Proposition 7 together with the objective attained by the geometric ground truth $\mathbf{P}_{\mathrm{gt}}$. The ground truth reaches between 91.1% and 99.8% of $\mathrm{UB}_{\mathrm{LAP}}$, with a mean of 95.6%, so on real data $\mathrm{UB}_{\mathrm{LAP}}$ lies within a few percent of the value that a correct solution attains, and it is tighter than $\mathrm{UB}_{\mathrm{spec}}$ on every scene. The residual gap is largest on *kicker*, *playground* and *courtyard*, the scenes with the noisiest observations, where part of the bound can be reached only by reproducing wrong observed matches. This is also why, on noisy $\mathbf{W}$, a higher objective does not by itself mean a more accurate matching once it exceeds the level of the ground truth, and why the initialization of Appendix E is chosen by cycle consistency and not by objective value.

C Compared Methods

This appendix describes what each compared method does, together with the weakness that motivates our approach.

Spectral (2013) [16] relaxes the permutation constraints to an orthogonality constraint and takes the leading eigenvectors of the observation matrix as a continuous universe embedding, which is then rounded to permutations. The relaxation loses tightness, and the method presumes full bijective correspondences, so partial matches are recovered only through the final rounding.

NmfSync (2019) [36] synchronizes partial matchings through a non-negative matrix factorization of the observation matrix, followed by a projection of the factor onto partial permutations. The

Table S1: The two upper bounds and the objective attained by the geometric ground truth on every scene.

Scene	$\mathrm{UB}_{\mathrm{LAP}}$	$\mathrm{UB}_{\mathrm{spec}}$	$f(\mathbf{P}_{\mathrm{gt}})$	$f(\mathbf{P}_{\mathrm{gt}})/\mathrm{UB}_{\mathrm{LAP}}$ (%)
terrace	1504	1658	1418	94.3
office	1478	1618	1410	95.4
electro	1706	1753	1668	97.8
kicker	1398	1625	1273	91.1
terrains	1674	1754	1650	98.6
relief	1828	1875	1791	98.0
relief_2	1726	1795	1672	96.9
courtyard	1220	1431	1124	92.1
delivery_area	1736	1811	1667	96.0
pipes	1404	1560	1351	96.2
playground	1380	1615	1260	91.3
facade	1582	1710	1579	99.8

factorization is solved in a continuous domain, and the projection that produces the output is not part of the optimized criterion.

Stiefel (2021) [37] maximizes the trace objective over the Stiefel manifold with an orthogonal iteration augmented by a sparsity-promoting step. The orthogonality constraint is a relaxation (in the sense of losing tightness), the resulting sparsity is only approximate, and a binary matching is obtained only after rounding.

FCC (2021) [38] filters the observed matches by cycle-consistency statistics computed over the match graph and keeps those whose statistics exceed a threshold. Its thresholded output can assign one keypoint to several partners and is therefore not a matching. We report it after the same exact block LAP applied to every other method, which yields a valid partial permutation.

MatchFAME (2022) [39] estimates the corruption level of every observed match by cycle-edge message passing and then runs a weighted projected power method. Its accuracy is decided by the corruption estimates and by the projection step, neither of which maximizes the synchronization objective.

GM-SEQSEQ and GM-PARPAR (2024) [8] build a multi-matching by a sequential or a parallel construction and refine it with a graph-matching local search. **GREEDA (2025)** [40] alternates this local search with a swap-based local search until neither improves the solution. These solvers carry no universe variable and no convergence guarantee, and they assign every keypoint, including distractors that have no correct partner.

D Additional Results

Tables S2–S4 report the ten scenes not shown in the main paper. The protocol is identical for every scene and every method, all methods receive the identical observations, and the runtimes of initialization-based methods include their initialization.

Reading of the per-scene results. Our method attains the best F-score and the best precision on every scene except *electro*. The margin over the strongest baseline is largest where the observations are noisiest or the structure most repetitive, 7.0 points of F-score on *courtyard*, 4.2 on *terrace*, 3.8 on *delivery_area* and 3.7 on *facade*, and it reaches 15.6 points over MatchFAME, the starting point of our ascent, on *kicker*. These are the scenes on which a matching decided by rounding, thresholding or local exchanges makes the most mistakes, and on which an exact update that gives each keypoint the label with the most support across the collection pays off most. On the least noisy scenes, *relief*, *relief_2* and *electro*, the observations are already largely correct, the universe-based methods lie within a few points of one another, and our method is on par with the strongest of them. The local-search solvers trail on every scene in precision, by 13 to 23 points against ours, because they assign every keypoint, including the distractors that have no correct partner. Our runtime stays

Table S2: Results on the ETH3D sequences *terrace*, *electro*, *kicker* and *terrains*, with the best value in bold and the second best underlined.

Dataset	Method	F-score (%) ↑	Precision (%) ↑	Cyclic (%) ↑	Runtime (s) ↓
terrace	Spectral (2013) [16]	<u>72.5</u>	<u>72.6</u>	**100.0**	0.075
	NmfSync (2019) [36]	72.1	72.2	**100.0**	0.242
	Stiefel (2021) [37]	70.1	70.2	**100.0**	0.405
	FCC + exact rounding (2021) [38]	69.8	69.9	**100.0**	0.180
	MatchFAME (2022) [39]	69.4	69.4	**100.0**	**0.041**
	GM-SEQSEQ (2024) [8]	66.7	59.0	**100.0**	0.316
	GM-PARPAR (2024) [8]	67.5	59.7	**100.0**	0.372
	GREEDA (2025) [40]	67.0	59.3	**100.0**	0.646
	Ours	**76.7**	**76.7**	**100.0**	<u>0.043</u>
electro	Spectral (2013) [16]	92.3	92.3	**100.0**	0.074
	NmfSync (2019) [36]	**92.9**	**92.9**	**100.0**	0.237
	Stiefel (2021) [37]	90.9	90.9	**100.0**	0.195
	FCC + exact rounding (2021) [38]	<u>92.4</u>	<u>92.4</u>	**100.0**	0.180
	MatchFAME (2022) [39]	89.3	89.3	**100.0**	**0.036**
	GM-SEQSEQ (2024) [8]	81.9	72.5	**100.0**	0.320
	GM-PARPAR (2024) [8]	81.4	72.0	**100.0**	0.343
	GREEDA (2025) [40]	82.1	72.6	**100.0**	0.638
	Ours	<u>92.4</u>	<u>92.4</u>	**100.0**	<u>0.056</u>
kicker	Spectral (2013) [16]	60.0	61.2	**100.0**	0.075
	NmfSync (2019) [36]	<u>64.1</u>	<u>65.3</u>	**100.0**	0.258
	Stiefel (2021) [37]	63.8	65.0	**100.0**	0.285
	FCC + exact rounding (2021) [38]	62.3	63.5	**100.0**	0.183
	MatchFAME (2022) [39]	51.3	52.3	**100.0**	**0.045**
	GM-SEQSEQ (2024) [8]	58.3	52.1	99.9	0.345
	GM-PARPAR (2024) [8]	58.9	52.7	99.9	0.419
	GREEDA (2025) [40]	60.1	53.7	99.9	0.689
	Ours	**66.9**	**68.2**	**100.0**	<u>0.054</u>
terrains	Spectral (2013) [16]	80.7	81.1	**100.0**	0.060
	NmfSync (2019) [36]	81.0	81.5	**100.0**	0.252
	Stiefel (2021) [37]	78.9	79.3	**100.0**	0.270
	FCC + exact rounding (2021) [38]	78.2	78.6	**100.0**	0.181
	MatchFAME (2022) [39]	<u>84.9</u>	<u>85.4</u>	**100.0**	**0.041**
	GM-SEQSEQ (2024) [8]	75.4	67.0	99.9	0.320
	GM-PARPAR (2024) [8]	74.1	65.8	99.9	0.331
	GREEDA (2025) [40]	75.8	67.4	99.9	0.470
	Ours	**86.5**	**87.0**	**100.0**	<u>0.050</u>

Table S3: Results on the ETH3D sequences *relief*, *relief_2*, *courtyard* and *delivery_area*, with the best value in bold and the second best underlined.

Dataset	Method	F-score (%) ↑	Precision (%) ↑	Cyclic (%) ↑	Runtime (s) ↓
relief	Spectral (2013) [16]	92.6	93.4	**100.0**	0.075
	NmfSync (2019) [36]	92.8	93.6	**100.0**	0.252
	Stiefel (2021) [37]	90.6	91.3	**100.0**	0.192
	FCC + exact rounding (2021) [38]	92.7	93.5	**100.0**	0.195
	MatchFAME (2022) [39]	<u>93.9</u>	<u>94.7</u>	**100.0**	**0.045**
	GM-SEQSEQ (2024) [8]	81.5	72.5	99.8	0.321
	GM-PARPAR (2024) [8]	82.7	73.6	99.8	0.410
	GREEDA (2025) [40]	82.0	72.9	99.8	0.666
	Ours	**94.7**	**95.5**	**100.0**	<u>0.050</u>
relief_2	Spectral (2013) [16]	89.0	89.4	**100.0**	0.074
	NmfSync (2019) [36]	88.0	88.5	**100.0**	0.254
	Stiefel (2021) [37]	88.3	88.7	**100.0**	0.195
	FCC + exact rounding (2021) [38]	89.6	90.1	**100.0**	0.190
	MatchFAME (2022) [39]	<u>90.6</u>	<u>91.1</u>	**100.0**	**0.048**
	GM-SEQSEQ (2024) [8]	80.8	71.8	99.9	0.332
	GM-PARPAR (2024) [8]	80.7	71.7	99.9	0.378
	GREEDA (2025) [40]	80.4	71.4	99.9	0.483
	Ours	**92.3**	**92.8**	**100.0**	<u>0.052</u>
courtyard	Spectral (2013) [16]	53.9	54.2	**100.0**	0.074
	NmfSync (2019) [36]	54.2	54.4	**100.0**	0.250
	Stiefel (2021) [37]	53.4	53.7	**100.0**	0.349
	FCC + exact rounding (2021) [38]	55.2	55.5	**100.0**	0.168
	MatchFAME (2022) [39]	<u>56.1</u>	<u>56.5</u>	**100.0**	**0.045**
	GM-SEQSEQ (2024) [8]	51.9	46.1	**100.0**	0.329
	GM-PARPAR (2024) [8]	53.5	47.5	**100.0**	0.382
	GREEDA (2025) [40]	53.8	47.8	99.7	0.673
	Ours	**63.1**	**63.4**	**100.0**	<u>0.060</u>
delivery_area	Spectral (2013) [16]	83.2	83.4	**100.0**	0.074
	NmfSync (2019) [36]	83.2	83.5	**100.0**	0.253
	Stiefel (2021) [37]	82.0	82.3	**100.0**	0.285
	FCC + exact rounding (2021) [38]	83.0	83.3	**100.0**	0.181
	MatchFAME (2022) [39]	<u>84.7</u>	<u>84.9</u>	**100.0**	**0.043**
	GM-SEQSEQ (2024) [8]	76.3	67.6	**100.0**	0.333
	GM-PARPAR (2024) [8]	77.6	68.8	**100.0**	0.373
	GREEDA (2025) [40]	77.6	68.8	**100.0**	0.485
	Ours	**88.5**	**88.7**	**100.0**	<u>0.047</u>

Table S4: Results on the ETH3D sequences *playground* and *facade*, with the best value in bold and the second best underlined.

Dataset	Method	F-score (%) ↑	Precision (%) ↑	Cyclic (%) ↑	Runtime (s) ↓
playground	Spectral (2013) [16]	59.6	60.3	**100.0**	0.075
	NmfSync (2019) [36]	61.3	62.0	**100.0**	0.250
	Stiefel (2021) [37]	61.8	62.5	**100.0**	0.322
	FCC + exact rounding (2021) [38]	61.1	61.8	**100.0**	0.166
	MatchFAME (2022) [39]	<u>62.8</u>	<u>63.4</u>	**100.0**	**0.040**
	GM-SEQSEQ (2024) [8]	56.7	50.5	99.8	0.323
	GM-PARPAR (2024) [8]	56.4	50.2	99.8	0.397
	GREEDA (2025) [40]	55.7	49.6	99.8	0.483
	Ours	**63.8**	**64.5**	**100.0**	<u>0.054</u>
facade	Spectral (2013) [16]	77.5	80.0	**100.0**	0.075
	NmfSync (2019) [36]	78.6	81.1	**100.0**	0.239
	Stiefel (2021) [37]	77.6	80.1	**100.0**	0.301
	FCC + exact rounding (2021) [38]	<u>78.9</u>	<u>81.5</u>	**100.0**	0.195
	MatchFAME (2022) [39]	72.2	74.5	**100.0**	**0.044**
	GM-SEQSEQ (2024) [8]	72.8	65.9	99.3	0.339
	GM-PARPAR (2024) [8]	72.6	65.7	99.2	0.399
	GREEDA (2025) [40]	72.5	65.6	99.2	0.674
	Ours	**82.6**	**85.3**	**100.0**	<u>0.056</u>

between 0.043 and 0.060 s on every scene, below that of every baseline except MatchFAME, whose output our ascent improves on every scene.

Selection of the displayed image pairs. For each scene, the pair shown in its qualitative figure is chosen among the ten image pairs by the following rule. A pair is eligible if, on that pair, our method has an F-score at least that of each of the eight displayed baselines, the fewest wrong matches of all nine methods, and at least 50 correct matches, so that the panel is readable. Among eligible pairs, the one with the largest F-score margin over the strongest displayed baseline is shown. The same solver outputs are used to select the pair and to draw it. Scores over all pairs of each scene are those of the tables above.

Reading of the qualitative results. Figures S1–S9 show the remaining scenes. On every displayed pair our method has the fewest wrong matches of the nine methods, for instance 3 wrong against at least 15 for every baseline on *courtyard*, 8 against at least 13 on *delivery_area*, and 22 against at least 27 on *terrace*. The local-search solvers recover as many correct matches as ours on several pairs, but always together with far more wrong ones, visible as long red diagonals that start at distractor keypoints. On *relief* and *electro*, whose observations are the cleanest, the displayed pair is one on which ours is on par with the strongest baseline, with 2 and 5 wrong matches respectively, and every universe-based method recovers a dense, largely correct set.

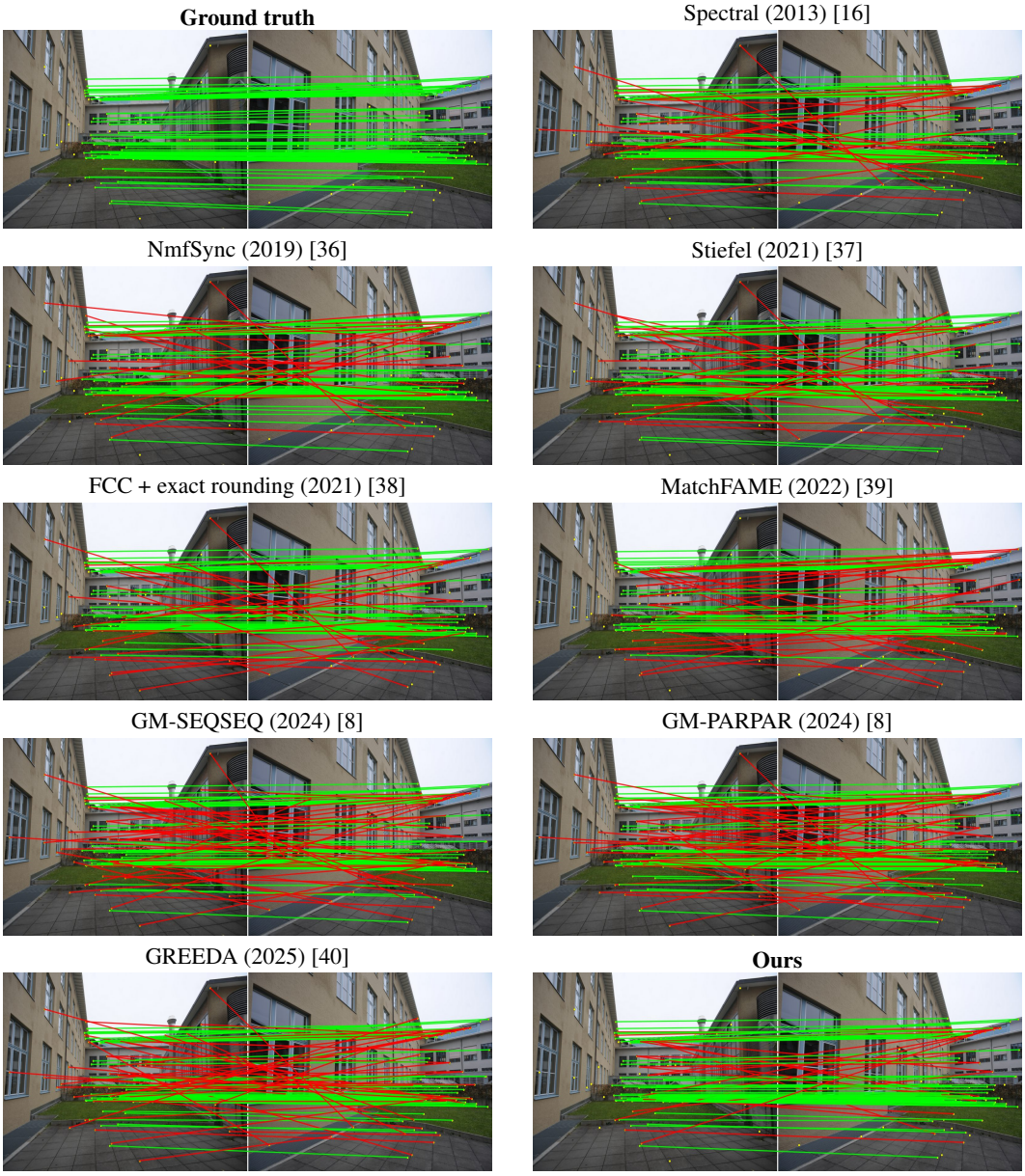

Figure S1: Correspondences recovered on one image pair of *terrace*. Green lines agree with the geometric ground truth, red lines are wrong, and yellow dots mark the keypoints.

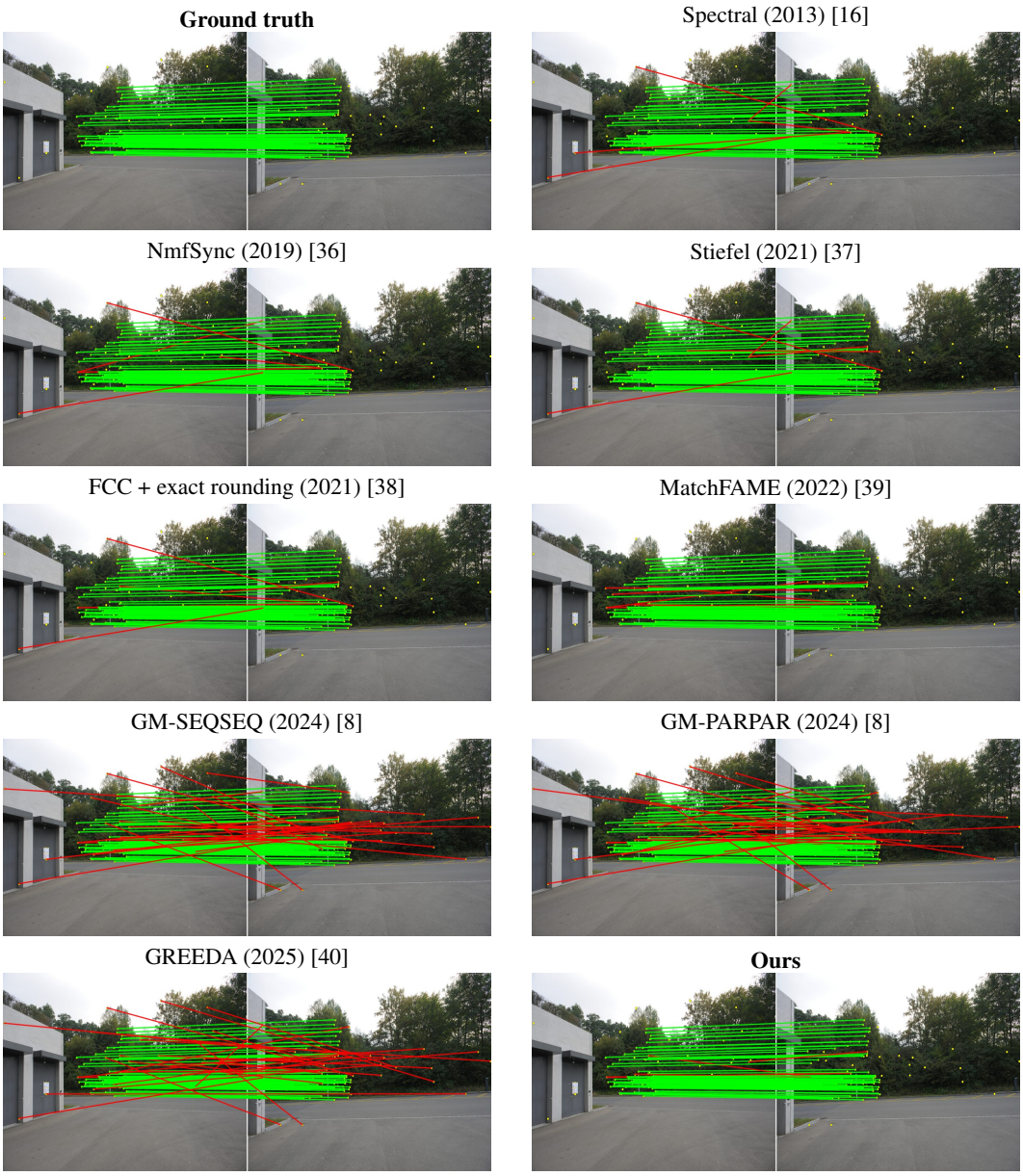

Figure S2: Correspondences recovered on one image pair of *electro*. Green lines agree with the geometric ground truth, red lines are wrong, and yellow dots mark the keypoints.

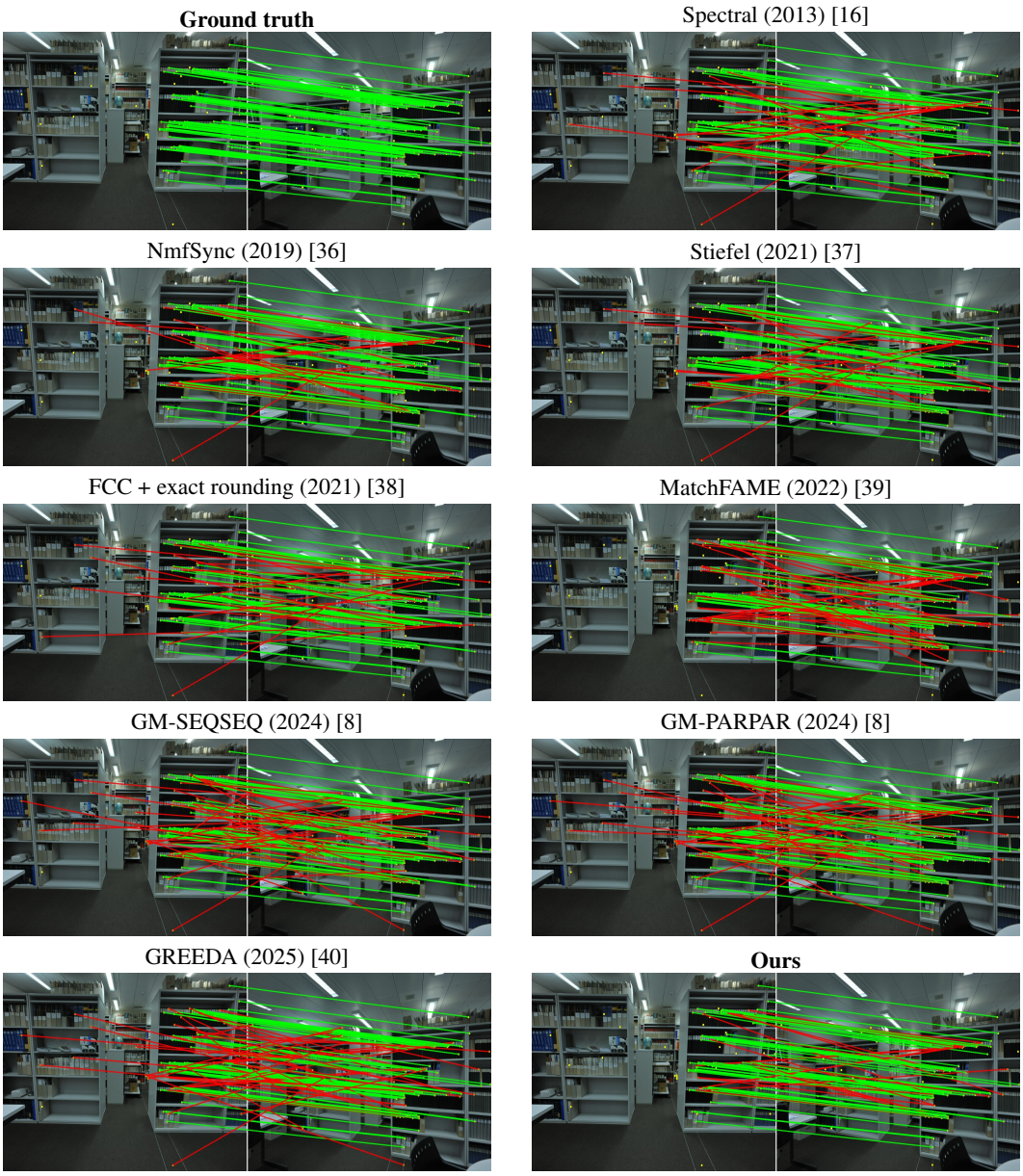

Figure S3: Correspondences recovered on one image pair of *kicker*. Green lines agree with the geometric ground truth, red lines are wrong, and yellow dots mark the keypoints.

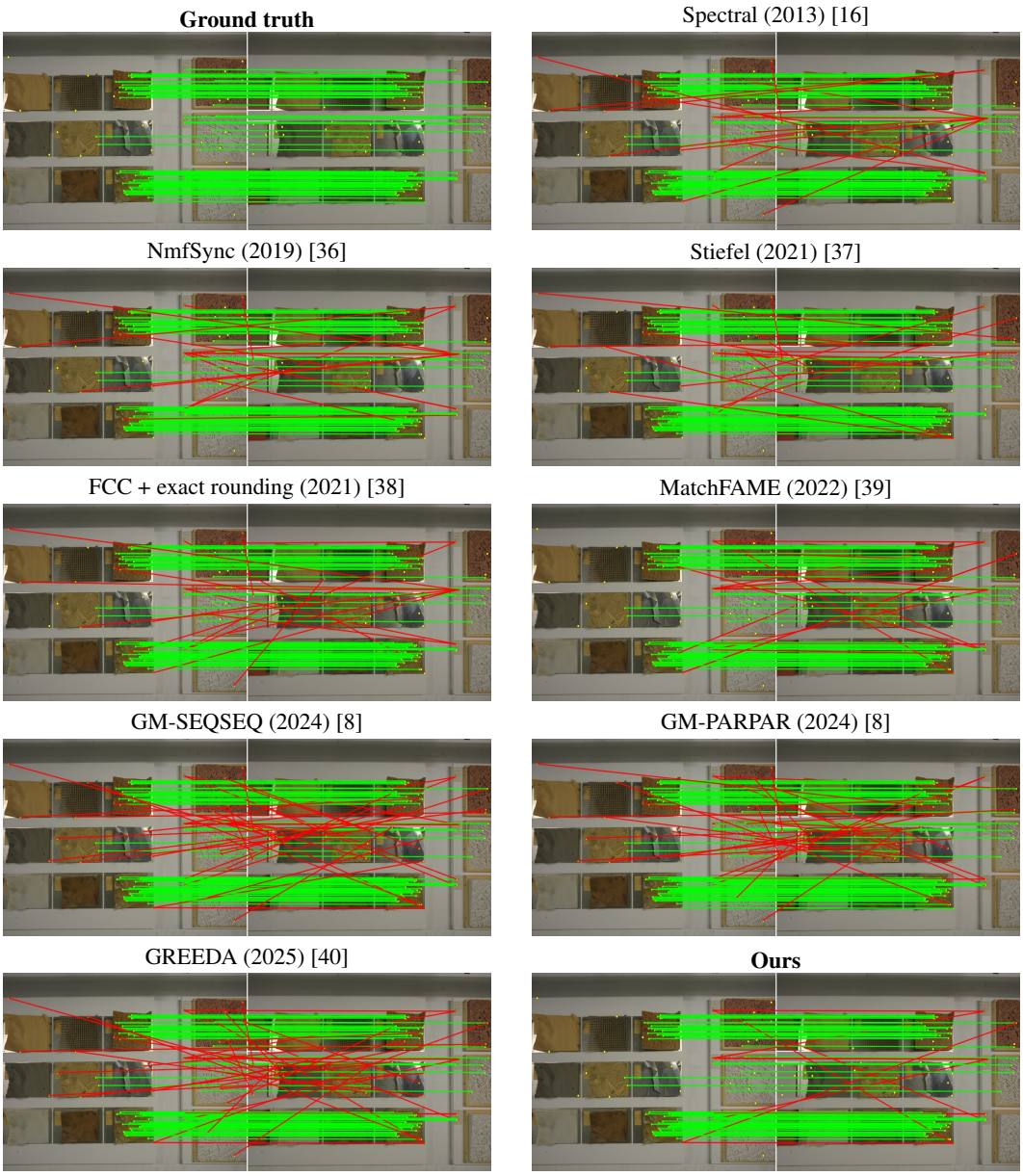

Figure S4: Correspondences recovered on one image pair of *terrains*. Green lines agree with the geometric ground truth, red lines are wrong, and yellow dots mark the keypoints.

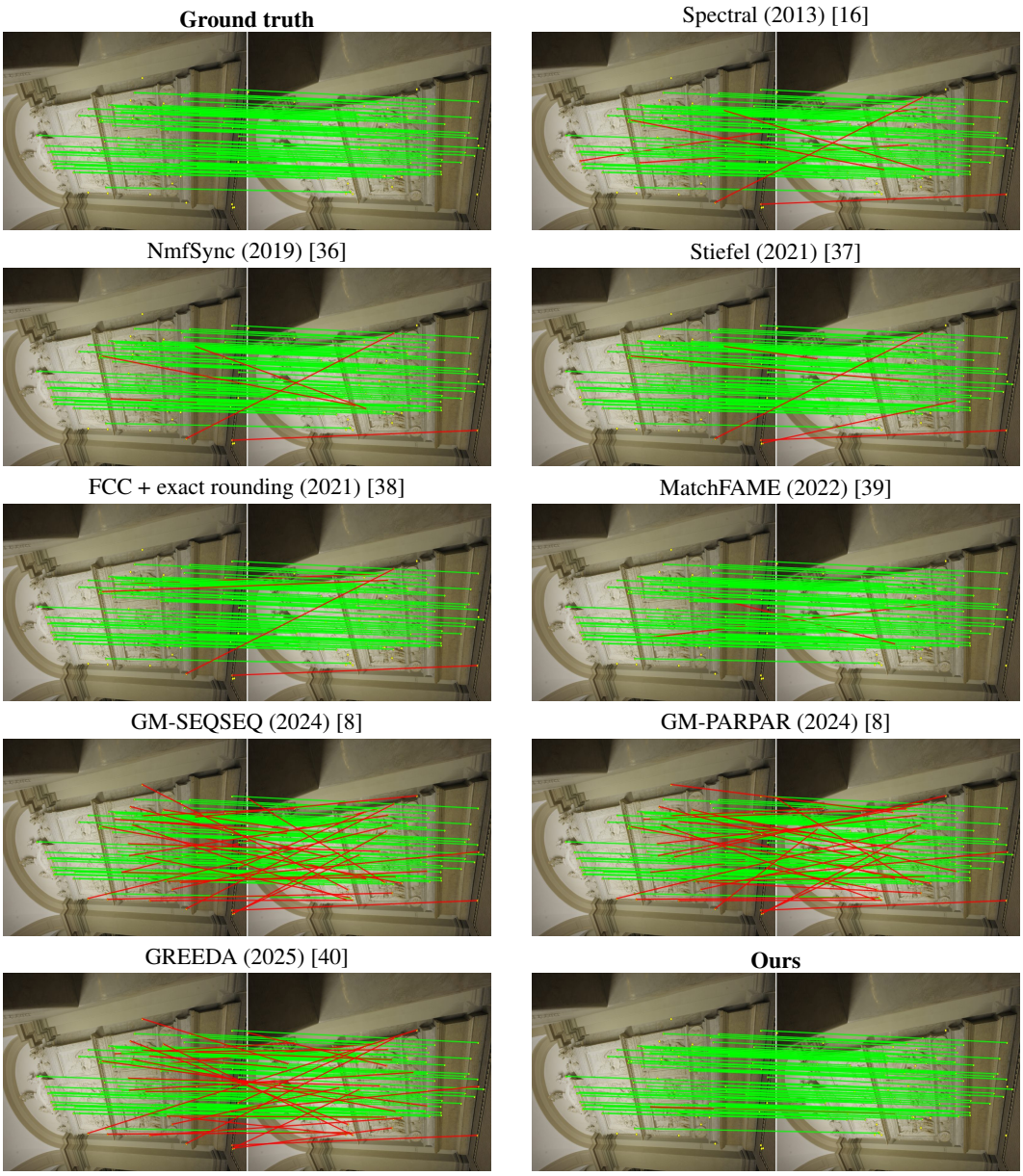

Figure S5: Correspondences recovered on one image pair of *relief*. Green lines agree with the geometric ground truth, red lines are wrong, and yellow dots mark the keypoints.

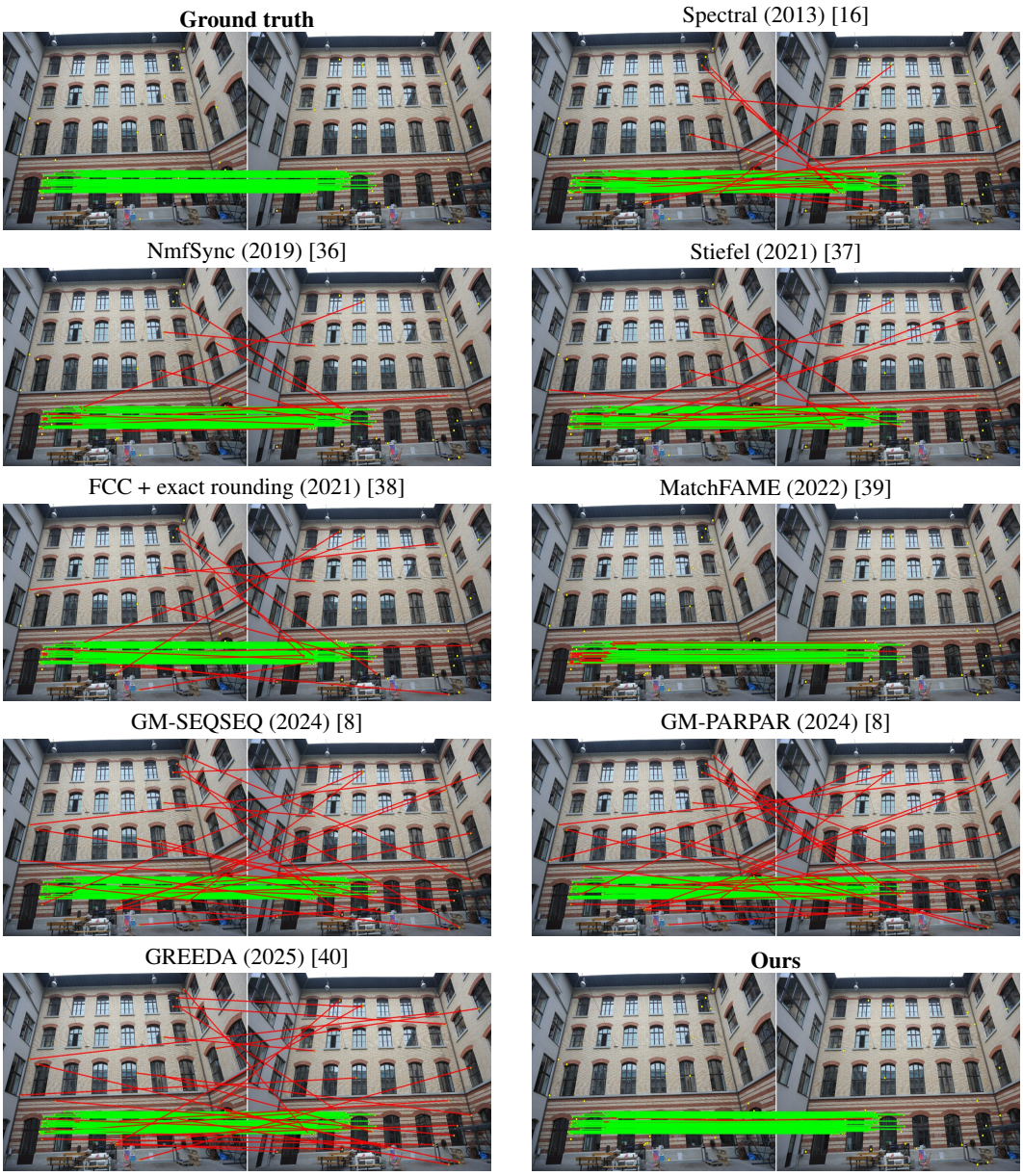

Figure S6: Correspondences recovered on one image pair of *courtyard*. Green lines agree with the geometric ground truth, red lines are wrong, and yellow dots mark the keypoints.

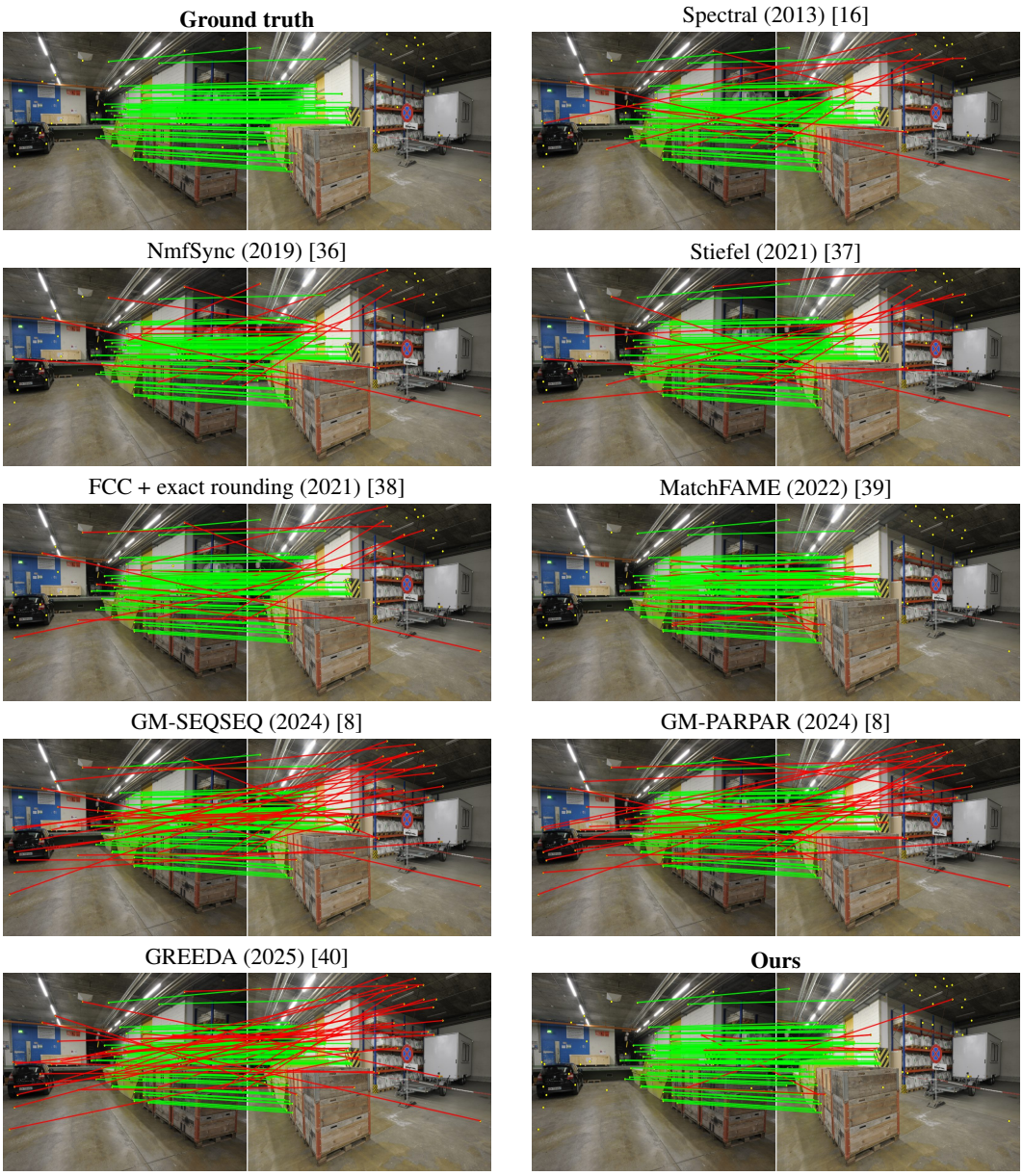

Figure S7: Correspondences recovered on one image pair of *delivery_area*. Green lines agree with the geometric ground truth, red lines are wrong, and yellow dots mark the keypoints.

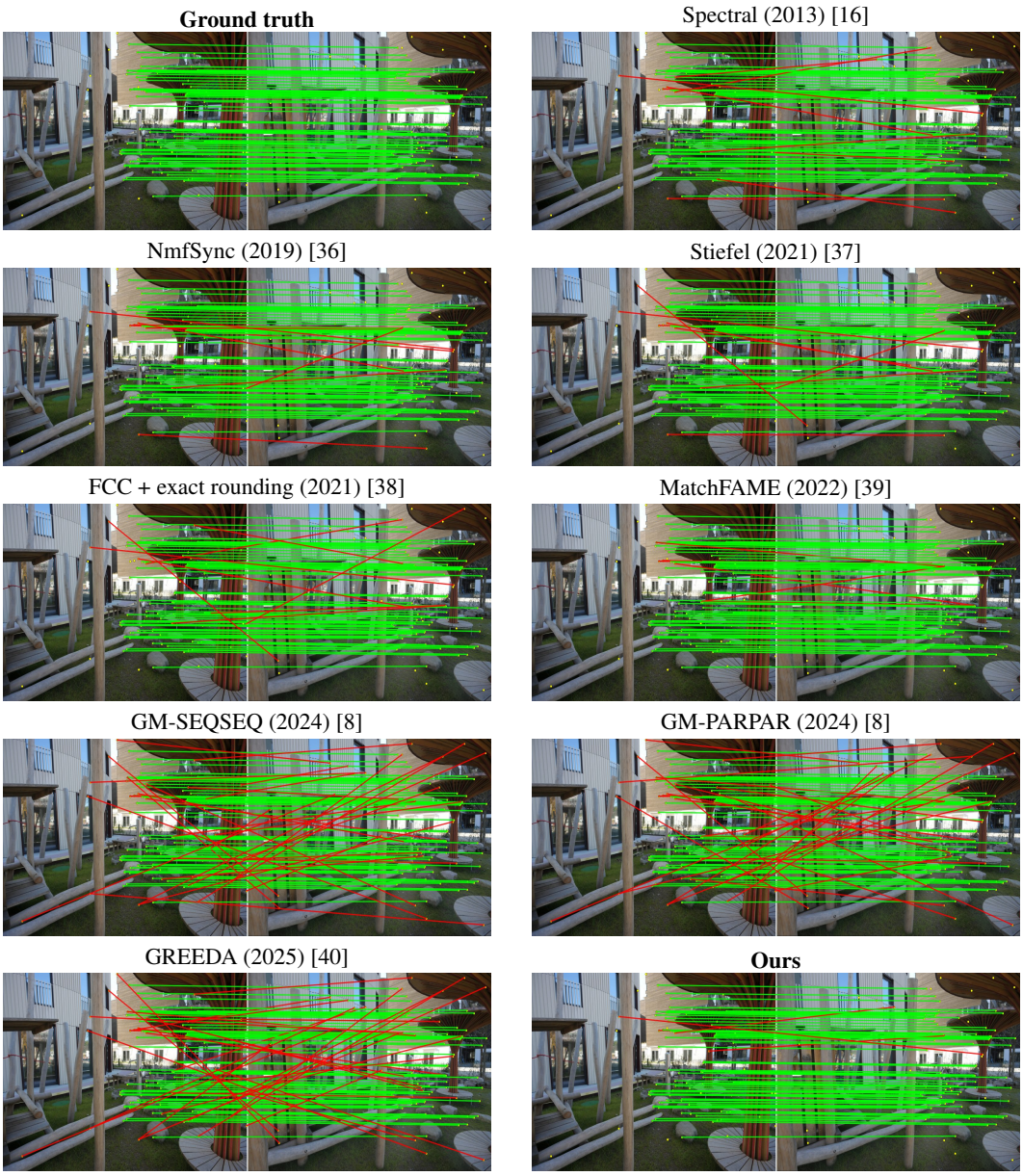

Figure S8: Correspondences recovered on one image pair of *playground*. Green lines agree with the geometric ground truth, red lines are wrong, and yellow dots mark the keypoints.

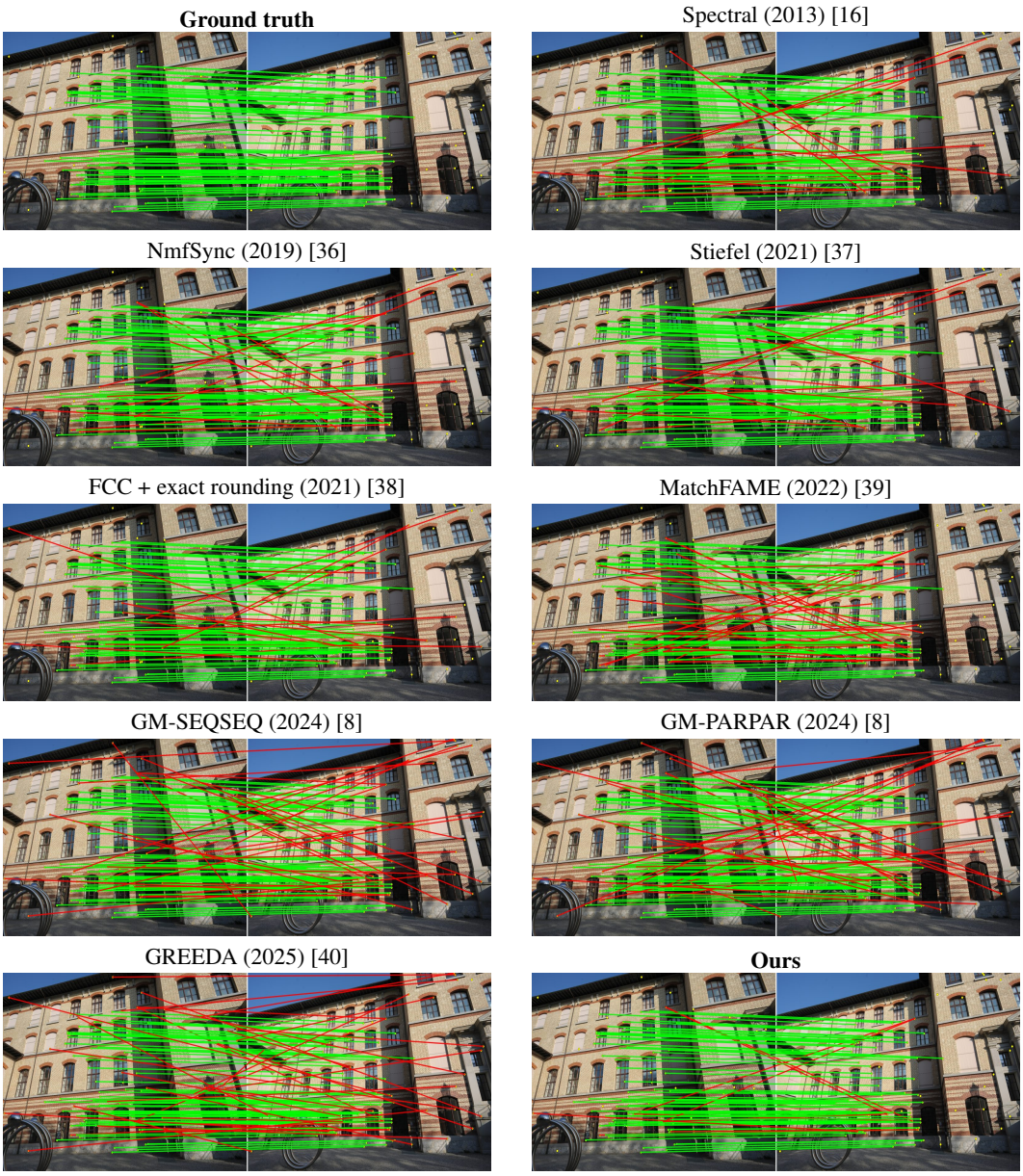

Figure S9: Correspondences recovered on one image pair of *facade*. Green lines agree with the geometric ground truth, red lines are wrong, and yellow dots mark the keypoints.

E Initialization

Our method starts from the assignment returned by MatchFAME [39], whose cycle-edge message passing estimates the corruption level of each observed match before committing to correspondences. This appendix reports how fast the ascent converges and how its result depends on the starting point.

Convergence. Figure S10 plots the objective after each pass on *terrace* and *electro*, started from MatchFAME, Stiefel, Spectral and random initializations. From every start the objective increases monotonically and reaches its final value within two to three passes, including random starts, whose objective climbs to the converged level in a single pass. This is the behaviour predicted by Proposition 4, and it shows that the worst-case bound on the number of passes is far from binding in practice.

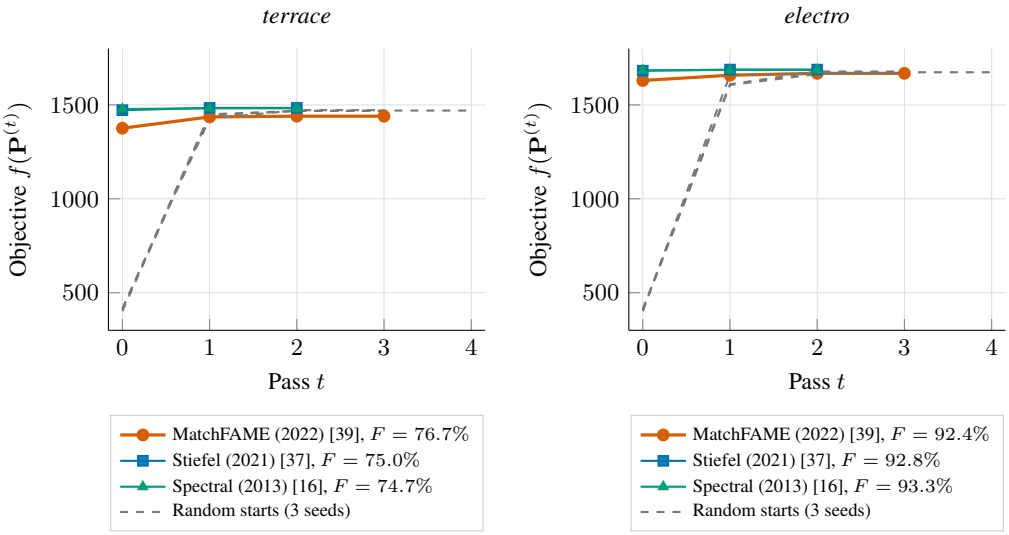

Figure S10: Objective after each pass of Algorithm 1 on *terrace* (left) and *electro* (right), from three structured initializations, with their final F-score in the legend, and from three random starts.

Refinement of the starting point. Table S5 compares the MatchFAME assignment alone with the result of our ascent started from it. The ascent raises the F-score on every scene, by 6.1 points on average and by up to 15.6 points on *kicker*, at an average additional cost of 10 ms. The gain is therefore produced by the exact updates, and not by the starting point.

Table S5: F-score (%) of the MatchFAME assignment alone and after our ascent started from it.

	terrace	*office*	*electro*	*kicker*	*terrains*	*relief*	*relief_2*	*courtyard*	*delivery_area*	*pipes*	*playground*	*facade*	**Mean**
MatchFAME (2022) [39] alone	69.4	67.1	89.3	51.3	84.9	93.9	90.6	56.1	84.7	65.9	62.8	72.2	74.0
After our ascent	**76.7**	**76.0**	**92.4**	**66.9**	**86.5**	**94.7**	**92.3**	**63.1**	**88.5**	**77.7**	**63.8**	**82.6**	**80.1**

Sensitivity to the initialization. Table S6 reports the final F-score reached from five initialization strategies on every scene. The ascent is stable, since random restarts vary by about one point of F-score within a scene, so a different start does not induce erratic behaviour and leads to a basin of comparable quality. The MatchFAME start gives the highest mean, 80.1%, about two points above the structured Stiefel, Spectral and NmfSync starts and above random starts, and it is the only start from which our method is ahead of every baseline on every scene except *electro*. This is

why we adopt it as the default. Its advantage agrees with Appendix B. On *terrace*, the Stiefel and Spectral starts lead to a slightly higher objective (Figure S10) yet to a lower F-score, because on noisy observations a higher objective beyond the level of the ground truth is bought by reproducing wrong observed matches. A start that is already selected for cycle consistency avoids that regime.

Table S6: Final F-score (%) after our ascent from five initializations (random is the mean $\pm$ standard deviation over 5 seeds).

Scene	**MatchFAME** (2022) [39]	**Stiefel** (2021) [37]	**Spectral** (2013) [16]	**NmfSync** (2019) [36]	**Random**
terrace	**76.7**	75.0	74.7	75.3	75.4 ± 0.5
office	**76.0**	71.2	71.9	71.1	72.3 ± 0.4
electro	92.4	92.8	**93.3**	**93.3**	92.1 ± 0.7
kicker	66.9	**67.4**	65.1	66.8	65.1 ± 2.3
terrains	**86.5**	82.7	82.9	82.9	84.3 ± 0.7
relief	**94.7**	93.9	93.9	93.2	94.3 ± 0.6
relief_2	**92.3**	90.0	90.7	90.6	90.3 ± 0.3
courtyard	**63.1**	57.8	57.4	57.6	57.9 ± 0.7
delivery_area	**88.5**	83.8	84.3	85.2	85.8 ± 0.7
pipes	**77.7**	75.7	77.4	76.7	77.0 ± 0.5
playground	**63.8**	62.3	62.3	62.2	63.1 ± 1.0
facade	**82.6**	81.8	81.3	**82.6**	81.9 ± 0.8
Mean	**80.1**	77.9	77.9	78.1	78.3

F Arbitrary Costs and Comparison on Common Objectives

This appendix shows that our method applies to cycle-consistent multi-matching with arbitrary costs, relates it to the multi-graph matching formulation of [8, 40], and compares the two on common objectives. It closes with further problems to which the same exact block update applies.

F.1 Arbitrary costs

Proposition 1 holds for any symmetric $\mathbf{W}$ and any $\mathbf{C}$, and it uses only that the constraints couple the variables within a single image. Nothing in it is specific to repairing a given set of observed matchings. Two cost types are of particular interest, both expressed through the recovered matching $\mathbf{W}^{\text{out}} = \mathbf{P}\mathbf{P}^T$.

(a) *Cycle-consistency repair with the fewest changes.* With $\mathbf{W}$ the binary matrix of observed matches, the cycle-consistent matching at minimum Hamming distance from $\mathbf{W}$ minimizes $\|\mathbf{W} - \mathbf{P}\mathbf{P}^T\|_F^2 = \|\mathbf{W}\|_F^2 - 2\langle \mathbf{W}, \mathbf{P}\mathbf{P}^T\rangle + \|\mathbf{P}\mathbf{P}^T\|_F^2$. On $\mathbb{U}$ the last term is constant, so the problem is equivalent to $\max_{\mathbf{P}\in\mathbb{U}} \operatorname{tr}(\mathbf{P}^T\mathbf{W}\mathbf{P})$, which is Problem (1).

(b) *Arbitrary linear costs.* Replacing $\mathbf{W}$ by an arbitrary symmetric affinity $\mathbf{A}$ gives $\max_{\mathbf{P}\in\mathbb{U}} \langle \mathbf{A}, \mathbf{P}\mathbf{P}^T\rangle$, and each block update remains an exact rectangular LAP by Proposition 1.

The multi-graph matching formulation of [8, 40] decomposes its cost over image pairs into vertex-to-vertex and vertex-pair-to-vertex-pair terms. Under $\mathbf{P}_{ij} = \mathbf{P}_i\mathbf{P}_j^T$, the vertex-to-vertex terms aggregate exactly into the cost of type (b), whereas the pair-to-pair terms are quartic in $\mathbf{P}$. That formulation carries no universe variable, since its solutions are pairwise matchings represented as cliques, and it is not given the number d of labels. Its per-image subproblem, which re-matches one image against the already matched remainder, is a quadratic assignment problem that is solved approximately, whereas the corresponding subproblem of our method is a linear assignment problem solved exactly.

F.2 A common energy and a common feasible set

We place both kinds of solvers on one objective, the agreement with the observed matches,

$$E(\mathbf{P}) = -\sum_{i<j} \langle \mathbf{W}_{ij}, \mathbf{P}_{ij}\rangle \qquad \text{(lower is better)}, \tag{S7}$$

which is the linear cost that the solver of [8, 40] optimizes on these instances, and which equals $-\frac{1}{2}\big(f(\mathbf{P}) - \sum_i \min(m_i, d)\big)$ on $\mathbb{U}$. Since that solver is not given d, its output uses about 105 clusters on scenes whose universe has 80 labels. To compare on one feasible set, we project its output into the d-label universe, grouping the keypoints into the connected components of the output matching, keeping at most one keypoint per image in each component (the one with the most matches inside it), and retaining the d largest components. Every method is run with its authors' solver on the identical twelve instances of Section 4, and (S7) is evaluated identically for all of them. Table S7 reports the result.

Table S7: The common energy (S7) on the d-label feasible set (lower is better), with the best value of each column in bold and the second best underlined.

Method	*terrace*	*office*	*electro*	*kicker*	*terrains*	*relief*
Ground truth	−509	−504	−634	−433	−624	−694
GM-SEQSEQ (2024) [8]	<u>−520</u>	<u>−498</u>	−632	−420	−609	<u>−691</u>
GM-PARPAR (2024) [8]	−514	−493	−632	−423	**−619**	**−692**
GREEDA (2025) [40]	**−521**	−492	**−635**	<u>−427</u>	−609	**−692**
Ours	<u>−520</u>	**−509**	<u>−634</u>	**−434**	**−619**	<u>−691</u>
Method	*relief_2*	*courtyard*	*delivery_area*	*pipes*	*playground*	*facade*
Ground truth	−635	−361	−633	−475	−428	−584
GM-SEQSEQ (2024) [8]	**−632**	−370	<u>−623</u>	−469	−457	<u>−526</u>
GM-PARPAR (2024) [8]	**−632**	−369	−622	**−476**	<u>−461</u>	−522
GREEDA (2025) [40]	**−632**	<u>−372</u>	−622	−467	<u>−461</u>	−514
Ours	**−632**	**−374**	**−631**	<u>−474</u>	**−467**	**−564**

On the common feasible set our method attains the lowest mean energy, -545.8 against -537.0 to -537.9 for the three local-search solvers. It attains the lowest energy on most scenes, by as much as 38 units on *facade*, and on the remaining ones it is on par with the best solver, within one or two units. The advantage of an exact update is thus visible on the objective itself, once all methods are confined to the feasible set that the problem specifies.

The level of the ground truth. Table S8 compares each method with the energy of the geometric ground truth, whose mean over the twelve scenes is -542.8. Without projection, the local-search solvers drive the energy about 26 units below that of the true correspondences. A solution can score better than the ground truth only by agreeing with observed matches that the ground truth rejects, and these are, by definition, wrong matches. The additional clusters and the matched distractors are where this surplus comes from, and it is accompanied by a precision and an F-score far below ours. Our method settles within three units of the level of the ground truth, and it is also the most accurate. Ordering by objective beyond the level of the ground truth and ordering by accuracy are therefore not the same ordering. The value of an exact per-image update is not that it drives a noisy objective further down without limit, but that within the feasible set of the problem each step is solved to global optimality with no rounding stage, and the solution consequently settles at the level that the true correspondences occupy.

Table S8: Mean over the twelve scenes of the gap to the ground-truth energy (method minus ground truth) and of the F-score, with and without the projection into the d-label universe.

Method	**Own output**		**Projected to d labels**	
	Gap	F (%)	Gap	F (%)
GM-SEQSEQ (2024) [8]	-25.6	69.9	$+5.6$	79.1
GM-PARPAR (2024) [8]	-26.4	70.3	$+4.9$	79.7
GREEDA (2025) [40]	-27.3	70.2	$+5.8$	79.4
Ours	$\mathbf{-2.9}$	**80.1**	$\mathbf{-2.9}$	**80.1**

F.3 A linear and quadratic cost

The multi-graph matching formulation is built for costs with both a linear and a quadratic term, so we also compare on such a cost. For every image pair (i, j) the objective to be maximized is

$$R_{ij}(\mathbf{P}_{ij}) = \langle \mathbf{W}_{ij}, \mathbf{P}_{ij} \rangle + \sum_{(r_1,r_2)\in\mathcal{E}_i} \sum_{(c_1,c_2)\in\mathcal{E}_j} q\big((r_1,r_2),(c_1,c_2)\big) \times \big(\mathbf{P}_{ij}[r_1,c_1]\,\mathbf{P}_{ij}[r_2,c_2] + \mathbf{P}_{ij}[r_1,c_2]\,\mathbf{P}_{ij}[r_2,c_1]\big), \tag{S8}$$

summed over all pairs, where $\mathcal{E}_i$ is the 4-nearest-neighbour graph of the keypoints of image i. The linear term is the observed match evidence. The quadratic term is a distance-preservation reward, $q = \lambda \exp\big(-(\delta_i - \delta_j)^2/(\sigma\bar{\delta})^2\big)$, where $\delta_i = \|\mathbf{x}_{r_1} - \mathbf{x}_{r_2}\|$ and $\delta_j = \|\mathbf{y}_{c_1} - \mathbf{y}_{c_2}\|$ are the lengths of the two edges, $\mathbf{x}_r$ and $\mathbf{y}_c$ denoting the image coordinates of keypoint r of image i and of keypoint c of image j, and $\bar{\delta}$ is the mean edge length of the pair, with $\lambda = 0.25$ and $\sigma = 0.3$, so a pair of matches is rewarded when it preserves the distance between its two keypoints. The instance is given to the solver of [8, 40] in its own input format, with every cost equal to the corresponding reward with its sign reversed, and it is run unmodified at its default settings. Our method maximizes the same reward over $\mathbb{U}$. Inside each block update it linearizes the quadratic term at the current iterate, which gives a linear function of $\mathbf{P}_i$ whose maximizer is again an exact rectangular LAP, and it accepts the update only if the reward increases, so the ascent is monotone and no rounding is introduced. It starts from its solution to Problem (1), and its runtime includes that solution. Table S9 reports the result on the twelve scenes.

Under this cost our method attains the highest mean F-score, 87.3% against 81.7 to 82.0% for the local-search solvers, and the best F-score on most scenes, by about ten points on *electro*, *terrains*, *relief*, *relief_2*, *delivery_area*, *pipes* and *facade*. It needs 1.4 s per scene on average, against 96 to 251 s, because each of its per-image subproblems is an exact linear assignment problem, whereas each subproblem of the local-search solvers is a quadratic assignment problem. As in Section F.2, the local-search solvers drive the reward beyond that of the ground truth on every scene, to 945.4–945.9 against 907.9 on average, whereas ours does not exceed it. On *kicker*, *courtyard* and *playground*, the

Table S9: F-score (%) and runtime (s) under the linear and quadratic cost (S8), with the best value of each column in bold and the second best underlined.

Method	*terrace*	*office*	*electro*	*kicker*	*terrains*	*relief*	*relief_2*	*courtyard*	*delivery_area*	*pipes*	*playground*	*facade*	**Mean**
F-score (%) ↑													
GM-SEQSEQ (2024) [8]	82.2	77.0	86.6	80.5	79.8	86.3	85.3	80.3	82.0	79.0	**80.7**	80.7	81.7
GM-PARPAR (2024) [8]	80.7	77.9	86.3	**80.8**	79.6	85.1	85.8	**82.6**	82.4	80.4	80.2	81.6	82.0
GREEDA (2025) [40]	80.4	76.7	87.1	80.2	80.5	85.0	85.4	80.7	82.0	81.5	80.6	80.6	81.7
Ours	**85.4**	**84.5**	**98.0**	78.6	**90.8**	**96.1**	**96.7**	76.7	**93.3**	**91.0**	64.9	**91.8**	**87.3**
Runtime (s) ↓													
GM-SEQSEQ (2024) [8]	78.8	91.3	107.6	168.7	113.7	59.5	82.4	59.9	80.7	111.2	115.3	82.1	95.9
GM-PARPAR (2024) [8]	153.2	171.0	248.0	223.7	215.2	258.0	353.4	260.1	193.1	249.4	390.7	292.8	250.7
GREEDA (2025) [40]	160.5	60.8	61.1	244.0	99.2	133.9	120.8	104.5	66.0	122.1	321.1	91.8	132.1
Ours	**1.7**	**1.4**	**1.2**	**1.9**	**1.2**	**1.2**	**1.2**	**1.4**	**1.2**	**1.6**	**1.3**	**1.3**	**1.4**

scenes with the noisiest observations, the local-search solvers reach a higher F-score under this cost, for which their model is designed, while our ascent, which linearizes the pair-to-pair term inside each exact block update, settles below the reward of the ground truth.

F.4 Further problems

The exact block update applies wherever the variables form a product of assignment sets and the objective is quadratic, and three such problems are worth mentioning. In multi-model fitting, noisy data points are grouped into several geometric structures while outliers are rejected, and recent formulations map this consensus problem to binary models solved on quantum annealers, which are limited by the available hardware and by embedding constraints [45]. Treating the assignment of points to models as a discrete assignment problem, our update solves each such subproblem exactly on classical hardware. In graph-based semi-supervised learning, labels are propagated from a few labelled examples through the graph structure, and standard techniques relax the discrete class indicators into continuous values, losing tightness, whose mapping back to classes introduces ambiguity [46]. Optimizing the label assignment directly keeps every inferred label a valid class throughout. Quadratic unconstrained binary optimization covers problems from Max-Cut to maximum independent sets, and learning-based solvers for it require extensive training data and generalize poorly to graph structures unseen during training [47]. A block update built on an exact linear subproblem requires no training and yields a deterministic, monotone ascent.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state that we keep permutation synchronization in its original combinatorial form and show that the per-image restriction of the objective is exactly linear (Lemma 3.1, Proposition 1); that each update is a rectangular linear assignment problem whose linear programming relaxation is tight by total unimodularity (Lemma 3.3, Proposition 2); that the method ascends monotonically and terminates after finitely many updates, with a block-wise optimal limit when ties are resolved in favour of the current assignment (Proposition 4); and that computable upper bounds on the optimum exist (Proposition 7). The empirical claims are supported by per-scene results on the twelve ETH3D multi-view sequences, with ground truth derived from calibrated 3D geometry, against eight baselines in F-score, precision, cyclic consistency and runtime (Table 1 and Appendix D).

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: A dedicated Limitations section (Section 5) immediately precedes the conclusion.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.

- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Every lemma, proposition and theorem states all of its assumptions, and complete proofs are given in Appendix A.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The datasets are the publicly available ETH3D high-resolution multi-view training sequences. Section 4 specifies how keypoints and ground truth are derived from the calibrated reconstruction, how the noisy observations are built (SIFT descriptors, nearest neighbours with the ratio test, single-view distractors), the number of views, and that all methods receive identical inputs. Algorithm 1 gives the method with its update, initialization and stopping rule, and the rule that selects each displayed image pair is stated in Appendix D.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example

(a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
(b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The code of our method, the processed inputs and the starting assignment of every scene, and the scripts that reproduce our results, the upper bounds and the convergence curves are provided as supplementary material, with a README giving the exact commands and environment. The raw ETH3D imagery is public and is downloaded from its official website.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 4 describes the data, the construction of the observations, the metrics and the initialization. Our method has no parameter to tune beyond the initialization, whose choice is studied in Appendix E. Each baseline is run at the parameter settings its own paper specifies.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Our method is deterministic given its input, and every method is evaluated on identical inputs, so the comparisons are paired. Results are reported separately for each of the twelve scenes (Table 1 and Appendix D), and the variability due to the initialization is reported as mean $\pm$ standard deviation over five random seeds (Appendix E).

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Section 4 states that all runs are performed on a machine with an AMD Ryzen 5 5600H CPU and 8 GB of RAM, and the runtime of every method is reported per scene. The HPC Center of Sharif University of Technology was used only for code development.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes]

Justification: We have reviewed the NeurIPS Code of Ethics, and our research conforms to it in every respect.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [N/A]

Justification: This is foundational algorithmic work on a combinatorial optimization problem, evaluated on a public benchmark of building and landscape imagery, with no direct path to a negative application.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: We release no model or dataset with a risk of misuse. The data we use are publicly available.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.

- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The ETH3D benchmark is cited and used under its Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License. Every baseline is cited, and each is run from the implementation its authors have published.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The new asset is our code together with the processed inputs derived from ETH3D, documented in a README that states the commands, the dependencies and the license, and provided as supplementary material. The processed inputs are shared under the license of ETH3D.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: Our work does not include crowdsourcing or research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: No human subject is involved in this work.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core method does not involve LLMs in any form. LLMs were used only for editing the text of the manuscript.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.