MPS: RELAXATION-FREE SOLVING FOR BLIND VISION–LANGUAGE MATCHING AT SCALE

Anonymous authors
Paper under double-blind review

ABSTRACT

In blind vision–language matching, the goal is to recover the correspondence between the class representations of a vision encoder and those of a language encoder without access to a single paired example. The task is naturally posed as the minimization of the Gromov–Wasserstein distortion between the intra-modal distance kernels of the two modalities, an instance of the quadratic assignment problem. We treat this problem entirely within its combinatorial domain, so that the search never leaves the set of permutations and no relaxation, in the sense of losing tightness, is introduced at any stage. Reducing the fourth-order Lawler formulation to an equivalent Koopmans–Beckmann trace maximization removes the cost tensor whose size otherwise limits the problems that can be posed. Shifting the spectrum of each kernel then makes the objective convex, and its minorization–maximization analysis shows why a single search trajectory halts short of the optimum, while a closed-form identity for the entire exchange neighbourhood, obtained from a single matrix product, makes a population-based search affordable at scale. We prove monotone ascent, convergence of the objective sequence and finite termination of the iterates. On benchmarks extending well beyond the scales previously studied, where the dual approach no longer delivers its guarantee of optimality, our solver outperforms state-of-the-art and benchmark methods in distortion at every problem size, matches more classes correctly than the dual solver on average, and remains interruptible at any moment with a feasible matching in hand.

1 INTRODUCTION

Vision and language models are trained on different data, with different objectives and different architectures. The platonic representation hypothesis nonetheless holds that, as models and their training corpora grow, the embeddings that different systems assign to the same underlying concepts induce increasingly similar patterns of pairwise similarity within each modality, so that two spaces may share structure even though neither was trained with knowledge of the other [13]. That structure has been exploited to connect unimodal encoders through a sparse set of anchors [28, 29]. It has also been studied through the lens of centered kernel alignment [16, 24]. The fully unsupervised setting removes even those anchors and seeks the correspondence between two sets of class representations when no example is paired [39]. The formulation is appealingly direct, in that one forms the matrix of pairwise distances between the class means within each modality, and then seeks the permutation that renders the two matrices as similar as possible. Measuring that similarity by the Gromov–Wasserstein distortion turns the search into a quadratic assignment problem, among the harder combinatorial problems to solve exactly even at moderate sizes [25, 15, 19].

The difficulty of the resulting optimization has been met from three broadly distinct directions, and it is worth separating them because each purchases something different. Primal heuristics search directly over permutations or over a continuous surrogate of them [46, 42, 8, 32, 7, 21]. Each of them follows a single trajectory and halts wherever that trajectory first ceases to improve, and the points at which it can halt are many and differ widely in quality. Convex relaxations, in the sense of losing tightness, instead embed the permutations into a larger set over which the problem becomes tractable, and then project back. As has been observed for semidefinite programs in other settings, dropping a rank constraint and reimposing it afterwards need not preserve tightness, so that the recovered point may be suboptimal even when the enlarged problem is solved exactly [2, 30].

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

The rounding that returns such a solution to a permutation is itself unaccounted for, and what it costs is not measured by the enlarged objective. Dual methods construct instead a lower bound on the optimum by Lagrangian ascent, and prove a feasible solution optimal once the bound meets its value [12]. That guarantee is bought at a cost per iteration growing as a high power of the number of classes, and at a memory footprint growing faster than the kernels themselves, both orders above what the primal route spends, and it holds only while the bound remains informative. The factorized ascent proposed with the unsupervised formulation leaves both the price and the condition in place, although it has made the dual route practical at moderate size. A fuller account of all three families is given in Section B.

In this work we keep the problem in its original combinatorial domain and solve it by a population-based primal search. Reducing the Lawler form of the objective to its Koopmans–Beckmann counterpart removes the fourth-order cost tensor whose size otherwise governs what can be posed, and shifting the spectrum of each kernel renders the reduced objective convex without disturbing any of its discrete optima. The minorization–maximization treatment of the shifted objective then reduces to a sequence of exact linear assignment problems, so that no rounding is required and the search never leaves the feasible set. The same analysis explains why a single trajectory of such an iteration comes to rest short of the optimum, and identifies breadth of search as the variable that matters. Our solver acts on this by maintaining a population of local optima of the exchange neighbourhood, every move of which is evaluated at once from a single matrix product, and by recombining them where they disagree. On benchmarks extending well beyond the sizes previously studied, the factorized dual solver proves optimality on no instance. Its lower bound never meets the attained value, turns non-positive and hence uninformative as the number of classes grows, and past a certain size the solver returns no solution within its budget. Over the same range our solver attains the lowest distortion of all methods evaluated at every problem size, matches more classes correctly than the dual solver on average, and returns a feasible matching throughout.

Main Contribution

The main contributions of this work are summarized as follows.

(i) **Direct combinatorial treatment.** We retain the problem in its original permutation domain, introducing no relaxation (in the sense of losing tightness), and show that shifting the spectrum of each kernel by its smallest eigenvalue yields an equivalent convex trace maximization over permutations, so that no heuristic rounding is required.

(ii) **A tensor-free formulation.** We pose the matching directly in its Koopmans–Beckmann form, so that the fourth-order cost tensor of the Lawler form is never instantiated and the problem carries only the two class kernels, which is what admits several hundred classes.

(iii) **Closed-form neighbourhood evaluation.** We establish an identity by which the change in the objective induced by every pairwise exchange is obtained simultaneously from a single dense matrix product, making a population-based search affordable at these scales.

(iv) **Monotonic ascent and convergence.** We prove that each iteration increases the objective monotonically, that the objective sequence is bounded above and below and therefore convergent, and that the iterates terminate finitely, at a point admitting no improving exchange in the case of the exchange search and at one optimal against its own linearization in the case of the iteration derived from minorization–maximization principles.

(v) **Superior solution quality at an anytime cost.** On standard image classification benchmarks and a zoo of vision encoders, our solver outperforms state-of-the-art and benchmark methods in distortion at every problem size measured and is more accurate than the dual solver on average, while a feasible matching is available whenever it is stopped.

(vi) **A characterization of the dual route beyond its studied range.** Over problem sizes extending well past those previously examined, we establish that the guarantee of optimality which motivates the dual formulation ceases to be delivered, that its lower bound degrades until it carries no information, and that past a certain size no solution is returned.

Notation. Boldface uppercase letters denote matrices and boldface lowercase letters vectors, $\mathbf{I}$ is the identity matrix and $\mathbf{1}$ the all-ones vector. We write N for the number of classes being matched, which we call the problem size, $\mathcal{P}_N$ for the set of $N \times N$ permutation matrices, and π for the permutation that a matrix $\mathbf{P} \in \mathcal{P}_N$ represents, with $\mathbf{P}^t$ the iterate at iteration t. The trace is $\mathrm{tr}(\cdot)$,

the smallest eigenvalue of a symmetric matrix is $\lambda_{\min}(\cdot)$, the Frobenius norm is $\|\cdot\|_F$ and the Frobenius inner product is $\langle \mathbf{A}, \mathbf{B} \rangle_F = \operatorname{tr}(\mathbf{A}^\top \mathbf{B})$. We write $\otimes$ for the Kronecker product, $\operatorname{vec}(\cdot)$ for column-wise vectorization, $\mathbf{Y}[\pi][:, \pi]$ for the matrix $\mathbf{Y}$ with its rows and columns reordered by π, and $\operatorname{Unif}(S)$ for the uniform distribution on a finite set S.

2 Problem Formulation

Consider a collection of N semantic classes observed independently through two modalities. On the vision side, each class i is represented by the renormalized mean $\mathbf{u}_i \in \mathbb{R}^{d_v}$ of the ℓ_2-normalized embeddings of its images, d_v being the dimension of the vision embedding; on the language side, by the renormalized mean $\mathbf{v}_j \in \mathbb{R}^{d_l}$ of the embeddings of its class name under a fixed collection of prompt templates, d_l being the dimension of the language embedding. Since the two embedding spaces are produced by encoders trained independently, with different objectives and of different dimension, no isometry between them may be assumed and their coordinates are not comparable. What can be compared is the *internal geometry* of each modality, and we therefore form the intra-modal kernels

$$X_{ik} = \big\|\mathbf{u}_i - \mathbf{u}_k\big\|_2, \qquad Y_{jl} = \big\|\mathbf{v}_j - \mathbf{v}_l\big\|_2, \qquad 1 \le i, k, j, l \le N, \tag{1}$$

so that $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{N \times N}$ are symmetric with zero diagonal, and blind matching becomes the problem of recovering the correspondence between the two class sets from (1) alone. Measuring the disagreement between the two geometries by the squared Gromov–Wasserstein distortion, and encoding the correspondence as a permutation matrix $\mathbf{P} \in \mathcal{P}_N$ in which $P_{ij} = 1$ precisely when vision class i is assigned to language class j, we seek the correspondence as the solution of

$$\min_{\mathbf{P} \in \mathcal{P}_N} F(\mathbf{P}) \triangleq \sum_{i,j,k,l=1}^{N} \big(X_{ik} - Y_{jl}\big)^2 P_{ij} P_{kl}. \tag{2}$$

The structure of (2) is worth reading before it is manipulated. The product $P_{ij}P_{kl}$ is unity exactly when the assignments $i \mapsto j$ and $k \mapsto l$ are made together, and $(X_{ik} - Y_{jl})^2$ is then the penalty incurred by that pair, being the squared discrepancy between the distance separating i from k in the vision geometry and that separating their images j and l in the language geometry. The objective accumulates such a penalty over every pair of assignments, so (2) is an instance of the Lawler form of the quadratic assignment problem, whose cost coefficients constitute an N^4 tensor. Solving the problem in this form would require the tensor itself, and the following lemma removes it.

Lemma 1 (Reduction to the Koopmans–Beckmann form)**.** *For symmetric* $\mathbf{X}$ *and* $\mathbf{Y}$ *and for all* $\mathbf{P} \in \mathcal{P}_N$*,*

$$F(\mathbf{P}) = \|\mathbf{X}\|_F^2 + \|\mathbf{Y}\|_F^2 - 2\,\operatorname{tr}\big(\mathbf{P}^\top \mathbf{X} \mathbf{P} \mathbf{Y}\big),$$

so that minimizing F *over* $\mathcal{P}_N$ *is equivalent to maximizing* $\operatorname{tr}(\mathbf{P}^\top \mathbf{X} \mathbf{P} \mathbf{Y})$ *over* $\mathcal{P}_N$ *(for the proof, see Section C.1).*

This identity is standard for the squared Gromov–Wasserstein cost, and what matters here is that it is applied before the fourth-order tensor is ever formed. Expanding the square separates the objective into three summations, of which the two carrying X_{ik}^2 and Y_{jl}^2 reduce to constants through the row and column sums of a permutation matrix; only the cross term depends on $\mathbf{P}$, and it reduces to $\operatorname{tr}(\mathbf{P}^\top \mathbf{X} \mathbf{P} \mathbf{Y})$. The derivation is given term by term in Section C.1. We therefore take as our working formulation

$$\max_{\mathbf{P} \in \mathcal{P}_N} f(\mathbf{P}) \triangleq \operatorname{tr}\big(\mathbf{P}^\top \mathbf{X} \mathbf{P} \mathbf{Y}\big), \tag{3}$$

the Koopmans–Beckmann form of the quadratic assignment problem. Two of its properties govern any algorithm that attacks it at scale. First, the problem is NP-hard and is among the hardest combinatorial problems to solve exactly even at moderate N [37]. Second, and of more practical consequence, Lemma 1 has eliminated the N^4 tensor, since a solver consuming (2) directly must materialize that tensor and is thereby subject to a memory ceiling of $O(N^4)$, whereas one operating on (3) escapes it entirely, which is what makes several hundred classes tractable in memory.

3 MPS: A POPULATION-BASED PRIMAL SOLVER

The problem (3) is a non-convex maximization over $\mathcal{P}_N$, whose cardinality $N!$ forbids enumeration beyond $N \approx 12$. To solve it we propose **MPS** (Memetic Primal Search), a population-based primal solver built on minorization–maximization principles (for an overview of MM, see Section A), which requires neither dual information, nor any enlargement of the feasible set, nor training, and which carries no tuned parameter beyond the population size. It rests on an analysis and two components. The analysis is an objective-preserving spectral shift which convexifies (3) and thereby identifies what the local step must supply (Section 3.1). The first component is an exact closed-form evaluation of the entire exchange neighbourhood at the cost of a single matrix product, which is what makes maintaining a population affordable (Section 3.2). The second is an agreement-preserving recombination operator that re-searches only the positions on which two local optima disagree (Section 3.3). For the convenience of the reader, the complete procedure is summarized in Section D.

3.1 CONVEXIFICATION BY SPECTRAL SHIFT

The objective (3) is a quadratic form in $\mathbf{P}$ and is in general neither convex nor concave, since the kernels (1) are Euclidean distance matrices and therefore indefinite. That indefiniteness can be removed without perturbing the discrete problem. We define the shifted kernels

$$\tilde{\mathbf{X}} = \mathbf{X} - \lambda_{\min}(\mathbf{X})\mathbf{I}, \qquad \tilde{\mathbf{Y}} = \mathbf{Y} - \lambda_{\min}(\mathbf{Y})\mathbf{I}, \tag{4}$$

both positive semidefinite by construction, and establish that the shift leaves the discrete problem untouched.

Proposition 1 (Invariance of the discrete optima)**.** *For every $\mathbf{P} \in \mathcal{P}_N$,*

$$\operatorname{tr}\big(\mathbf{P}^\top \tilde{\mathbf{X}}\mathbf{P}\tilde{\mathbf{Y}}\big) = \operatorname{tr}\big(\mathbf{P}^\top \mathbf{X}\mathbf{P}\mathbf{Y}\big) - \lambda_{\min}(\mathbf{Y})\operatorname{tr}(\mathbf{X}) - \lambda_{\min}(\mathbf{X})\operatorname{tr}(\mathbf{Y}) + \lambda_{\min}(\mathbf{X})\lambda_{\min}(\mathbf{Y})N,$$

so that the two objectives differ by an additive constant and are maximized by exactly the same permutations (for the proof, see Section C.2).

Lemma 2 (Convexity of the shifted objective)**.** *Let $\mathbf{p} = \operatorname{vec}(\mathbf{P})$. Then $\operatorname{tr}(\mathbf{P}^\top \tilde{\mathbf{X}}\mathbf{P}\tilde{\mathbf{Y}}) = \mathbf{p}^\top(\tilde{\mathbf{Y}} \otimes \tilde{\mathbf{X}})\mathbf{p}$, whose Hessian $2(\tilde{\mathbf{Y}} \otimes \tilde{\mathbf{X}})$ is positive semidefinite, so that the shifted objective is convex on $\mathbb{R}^{N\times N}$ (for the proof, see Section C.3).*

Two consequences follow, and together they determine the local move. The first concerns the feasible set, over which a convex function attains its maximum at a vertex, and the vertices of the Birkhoff polytope are precisely the elements of $\mathcal{P}_N$, so relaxing the integrality constraint does not change the optimal value, no tightness is lost, and no rounding is required at any stage. The second concerns the ascent direction. Writing $\tilde{f}(\mathbf{P}) = \operatorname{tr}(\mathbf{P}^\top \tilde{\mathbf{X}}\mathbf{P}\tilde{\mathbf{Y}})$, convexity implies that the first-order Taylor expansion of $\tilde{f}$ about the current iterate $\mathbf{P}^t$ is a global underestimator which is tight at that iterate, and is therefore a valid minorizer in the sense the MM principle requires,

$$g\big(\mathbf{P} \mid \mathbf{P}^t\big) = \tilde{f}\big(\mathbf{P}^t\big) + \big\langle \nabla\tilde{f}\big(\mathbf{P}^t\big), \mathbf{P} - \mathbf{P}^t\big\rangle_F \;\le\; \tilde{f}(\mathbf{P}), \qquad \forall\, \mathbf{P}, \tag{5}$$

with equality at $\mathbf{P} = \mathbf{P}^t$. Differentiating the quadratic form and using the symmetry of the shifted kernels gives $\nabla\tilde{f}(\mathbf{P}) = 2\,\tilde{\mathbf{X}}\mathbf{P}\tilde{\mathbf{Y}}$, and since $\tilde{f}(\mathbf{P}^t)$ and $\langle \nabla\tilde{f}(\mathbf{P}^t), \mathbf{P}^t\rangle_F$ are constants that may be discarded, maximizing the minorizer leaves

$$\mathbf{P}^{t+1} \in \arg\max_{\mathbf{P}\in\mathcal{P}_N} \operatorname{tr}\big((\mathbf{G}^t)^\top \mathbf{P}\big), \qquad \mathbf{G}^t = 2\tilde{\mathbf{X}}\mathbf{P}^t\tilde{\mathbf{Y}}, \tag{6}$$

where ties are broken in favour of $\mathbf{P}^t$, which ensures that the iterates, and not only their objective values, terminate. We refer to (6) as the linearization iteration. Its significance is that its objective is *linear* in $\mathbf{P}$, so that what began as a quadratic assignment problem has become, at each iteration, a linear assignment problem, solved exactly in polynomial time, $O(N^3)$, by the Hungarian algorithm [18]. The minorizer (5) guarantees that the objective does not decrease along the resulting sequence.

The derivation also locates the limitation shared by every scheme that follows a single trajectory of linear assignment steps, the Frank–Wolfe scheme FAQ among them. Such a scheme halts at a

permutation optimal against its own linearization, which need not be a global optimum, and since the maximum of a convex function is attained at a vertex, there is no interior stationary point to escape and no saddle structure to exploit. Those vertices are many and their objective values differ considerably, so a single run is limited by the one trajectory it follows, a limit that no finer local step removes. Breadth of search is accordingly the variable that matters, and MPS is built on this observation.

3.2 Exact Evaluation of the Exchange Neighbourhood

Maintaining a population is affordable only if the refinement applied to each member is cheap. The natural local move on $\mathcal{P}_N$ is the exchange of two assignments, and evaluating each of the $\binom{N}{2}$ such moves by recomputing the objective costs $O(N^2)$ per move and hence $O(N^4)$ per pass, which is prohibitive at the sizes we consider. For the factored objective (3) this cost admits a substantial reduction.

Lemma 3 (One-product neighbourhood evaluation)**.** *Let π be a permutation and let $\mathbf{B} = \mathbf{Y}[\pi][:, \pi]$ denote the language kernel reordered by π. For symmetric $\mathbf{X}$ and $\mathbf{B}$, the change in $\operatorname{tr}(\mathbf{P}^\top \mathbf{X}\mathbf{P}\mathbf{Y})$ induced by exchanging the assignments at positions a and b is given, for all pairs simultaneously, by*

$$\mathbf{\Delta} = 2\big(\mathbf{S} + \mathbf{S}^\top - \mathbf{r}\mathbf{1}^\top - \mathbf{1}\mathbf{r}^\top - \mathbf{K}^{(1)} - \mathbf{K}^{(2)}\big) + \mathbf{C},$$

where $\mathbf{S} = \mathbf{X}\mathbf{B}$, $r_i = \sum_j X_{ij}B_{ij}$, and $\mathbf{K}^{(1)}$, $\mathbf{K}^{(2)}$, $\mathbf{C}$ are corrections formed entrywise from the diagonals of the two kernels and their (a, b) entries, at $O(N^2)$ and no further matrix product (for the proof, see Section C.4).

The practical content of Lemma 3 is that the entire neighbourhood follows from the single matrix product $\mathbf{X}\mathbf{B}$, so that one pass costs $O(N^3)$ and every exchange is scored at once by one dense matrix multiplication. What is new is this simultaneous closed form, where the incremental delta evaluation of robust tabu search computes the change for each exchange separately. We adopt a best-improvement policy, each round scoring the whole neighbourhood and applying the single most improving exchange, since the deltas are exact only with respect to the current permutation and applying several interacting exchanges within one round makes the search oscillate.

3.3 Agreement-Preserving Recombination

The remaining component is the operator by which the population is refined. Let π_a and π_b be two distinct members, each a local optimum of the exchange search, and let $\mathcal{A} = \{\, i : \pi_a(i) = \pi_b(i) \,\}$ and $\mathcal{D} = \{1, \dots, N\} \setminus \mathcal{A}$ be the positions on which they agree and disagree. The operator produces a child π_c by fixing the assignments the two parents already share and re-searching only the remainder,

$$\pi_c(i) = \pi_a(i) \;\text{ for } i \in \mathcal{A}, \qquad \big(\pi_c(i)\big)_{i \in \mathcal{D}} \sim \operatorname{Unif}\big(\Pi(\mathcal{L}_\mathcal{D})\big), \tag{7}$$

where $\mathcal{L}_\mathcal{D}$ is the set of labels left unassigned by $\mathcal{A}$ and $\Pi(\cdot)$ denotes the set of its orderings. The child is then refined by the same local search and admitted into the population if it strictly improves upon the weakest member and is not already present.

The rationale follows from the analysis of Section 3.1. If a single trajectory halts at a vertex optimal only against its own linearization, then the informative signal lies where several independent trajectories stop in the same place. Positions on which two independent local optima agree are more likely to be shared with the optimum than positions on which they differ, which is the premise of memetic recombination for this problem [27, 26]. Restricting the re-search to $\mathcal{D}$ therefore concentrates the available effort where it can still alter the outcome. A child inherits its parents' assignments outside $\mathcal{D}$, so the number of refinement rounds it needs is governed by $|\mathcal{D}|$, which shrinks as the population converges and makes recombination progressively cheaper. What Lemma 3 adds to this template is that the refinement on which it depends remains affordable when N is in the hundreds.

4 Initialization, Computational Complexity and Convergence

4.1 Initialization

MPS is initialized from uniformly random permutations, and deliberately so. A structured initialization, such as a spectral or Sinkhorn-based starting point, would concentrate the population in a

single basin and thereby defeat the diversity on which (7) depends, since if all members agree almost everywhere then $\mathcal{D}$ is nearly empty and the operator degenerates into a copy. This reasoning is tested against four alternative starting points in Section F.1. We cap the population-construction phase at one half of the budget so that a badly conditioned instance cannot consume the whole allocation before a single recombination is attempted, and since a feasible permutation is held from the first iteration onward, a solution is always produced.

4.2 Computational Complexity

The per-operation costs follow from Lemma 3. One round of the exchange search, which is the local step of the solver we report, evaluates the whole neighbourhood through a single matrix product at $O(N^3)$, and so costs $O(N)$ per candidate move amortized over the $\binom{N}{2}$ it scores; recombination is $O(N)$, and a child needs a number of rounds governed by $|\mathcal{D}|$. Each step of the linearization iteration (6), measured as a baseline in Section F, costs one gradient and one linear assignment problem, each $O(N^3)$, after the one-off shift (4). The memory footprint is $O(N^2)$ throughout and the N^4 cost tensor of the Lawler form is never materialized, which is what admits problems of several hundred classes and stands in contrast to the $O(N^5)$ time and $O(N^3)$ memory per iteration of the dual approach.

4.3 Convergence

Proposition 2 (Monotonicity, boundedness and convergence)**.** *Let $\{\mathbf{P}^t\}$ be the sequence of iterates produced by either the exchange search or the linearization iteration (6), and $\{f(\mathbf{P}^t)\}$ the corresponding objective sequence. Then* (a) *$f(\mathbf{P}^{t+1}) \geq f(\mathbf{P}^t)$ for every t, with strict inequality at every accepted exchange;* (b) *$0 \leq f(\mathbf{P}^t) \leq \|\mathbf{X}\|_F \, \|\mathbf{Y}\|_F$, so the objective sequence is bounded above and below;* (c) *$\{f(\mathbf{P}^t)\}$ is monotone and bounded, hence convergent, and attains its limit after finitely many iterations; and* (d) *$\{\mathbf{P}^t\}$ terminates finitely, at a point admitting no improving exchange for the exchange search and at one optimal against its own linearization for (6) respectively, neither condition implying the other. Consequently the incumbent of MPS is non-decreasing in t and the procedure is anytime (for the proof, see Section C.5).*

5 Experimental Results

5.1 Compared Methods

We compare against state-of-the-art and benchmark methods from each of the three families of Section 1. *Random* assigns classes uniformly at random and fixes the level below which no method should be considered informative. Among the primal heuristics we evaluate *FAQ*, the Frank–Wolfe scheme over the Birkhoff polytope, which must round its iterate back to a permutation and follows one trajectory to a point optimal only against its own linearization; *GW-OT*, the entropic Gromov–Wasserstein solver built on the Sinkhorn iteration, where the entropic term perturbs the very objective that is solved and the coupling it returns is not a permutation until it too has been rounded; *2-opt*, the pairwise-exchange local search, which stops at the first permutation admitting no improving exchange and has no means of leaving it; and *BAPG*, a single-loop relaxation of the same objective, in the sense of losing tightness, which inherits the same rounding step. *LocalCKA*, the kernel-alignment method, coincides with FAQ on symmetric kernels and carries its limitation unchanged, as Section 5.4 shows. Finally, *FHG* is the factorized Hahn–Grant dual solver, whose guarantee of optimality costs $O(N^5)$ operations and $O(N^3)$ memory per iteration and holds only while its lower bound remains informative. A fuller account of each of these methods is given in Section B.

Fairness of the comparison. Every competing method is executed from the implementation released by its own authors and configured exactly as its own paper prescribes, and our own solver runs with one configuration throughout, a population of ten and no other tuned parameter. Since the dual solver is not an anytime method, we run it first, record the wall-clock time it consumes, and give our solver that same duration as its budget. Because a single iteration of an $O(N^5)$ ascent can outlast the budget, every solver is terminated externally at twice its nominal budget, and a run terminated in this way is recorded as returning no solution; where the dual solver is terminated so, ours receives the nominal budget for that size.

Table 1: Gromov–Wasserstein distortion (↓) at representative problem sizes. Shaded row is ours. **Bold** = best, underlined = second best.

	problem size N							
method	30	50	100	150	200	257	300	365
Random	119.1	331.9	1352.0	2485.7	4393.8	6473.9	11146.6	16544.1
FAQ (2015) [46]	115.3	320.4	1303.3	2386.7	4234.0	6356.6	10539.0	15691.9
GW-OT (2016) [32]	114.1	316.5	1283.5	2349.1	4163.8	6314.1	10220.4	15151.7
2-opt (2019) [8]	112.0	312.1	1259.6	2279.7	4045.8	6210.2	9918.6	—
BAPG (2023) [21]	112.9	312.1	1264.3	2303.3	4077.1	6251.9	10004.2	14761.7
LocalCKA (2024) [24]	115.3	320.4	1303.3	2386.7	4234.0	6356.6	10539.0	15691.9
FHG (2025) [39]	111.1	309.1	1253.7	2275.9	4033.3	6209.9	9925.4	—
MPS (ours)	**111.0**	**308.4**	**1249.2**	**2263.4**	**4015.5**	**6190.6**	**9826.4**	**14528.9**

5.2 Evaluation Metrics

Two quantities are reported. The first is the Gromov–Wasserstein distortion (2), the objective every solver optimizes and therefore the measure of optimization quality, which we recompute from the returned permutation so that differing offsets and sign conventions among the solvers' own reported values play no part. The second is the matching accuracy, the fraction of classes assigned to their true counterpart, read against the chance level $1/N$.

5.3 Evaluation Datasets and Experimental Protocol

For a dataset with N classes, the vision representation of a class is the renormalized mean of the ℓ_2-normalized embeddings of its images and the language representation the renormalized mean of the embeddings of its class name under the eighteen prompt templates accompanying the datasets, the two kernels being the matrices of pairwise Euclidean distances (1) within each modality. Since the problem sizes previously reported are bounded at one hundred classes by the datasets conventionally used, we evaluate on three datasets of increasing size, CIFAR-100, Caltech-256 and Places365, up to their complete class sets of one hundred, two hundred and fifty-seven and three hundred and sixty-five classes and without subsampling [17, 11, 47]. The encoders are the thirty-two vision and twenty-seven language encoders on which this task is conventionally studied. The comparison below uses six vision and two language encoders, all thirty-two vision encoders enter the zoo study of Section G.5, and Section E lists them. All results below are at $N \geq 30$, over six hundred and forty-two instances spanning thirty-two sizes.

5.4 Quantitative Performance Analysis

Table 1 reports the comparison at representative problem sizes, every size we evaluated being given in Section G.6. So that the columns are directly comparable, each is computed over the instances on which every solver evaluated at that size returned a solution, and a solver that returns nothing is recorded as absent. From the table we draw the following three observations.

Memetic primal search attains the lowest distortion at every problem size we measure. The ordering is stable across the range. Random assignment is worst throughout, FAQ and LocalCKA follow, then GW-OT and BAPG, then 2-opt and FHG, which change places with one another as the problem grows. Instance by instance, our solver attains a strictly lower distortion than 2-opt and BAPG on every instance on which both were run, and the mean excess of the competing methods over ours grows from a few percent for the strongest primal heuristics to above twenty percent for random assignment. FAQ and LocalCKA are identical at every size, since on symmetric kernels the two coincide by construction, and they should be read as one method. The advantage does not depend on the six vision encoders used here either, since over the thirty-two encoders of the zoo, at each dataset's complete class set, our solver attains the lowest distortion of every method run there on every encoder–dataset pair (Section G.5).

Table 2: Matching accuracy by band of problem size, in percent (↑), with the ratio to chance in parentheses. Shaded row is ours. **Bold** = best.

	band of problem size N			
	30–60	61–100	101–150	151–300
instances	192	204	120	96
chance $1/N$ (%)	2.45	1.20	0.78	0.46
FHG (2025) [39]	6.68 (2.73×)	2.84 (2.37×)	2.68 (3.45×)	2.45 (5.32×)
MPS (ours)	**7.21 (2.95×)**	**4.13 (3.45×)**	**3.37 (4.32×)**	**2.85 (6.20×)**

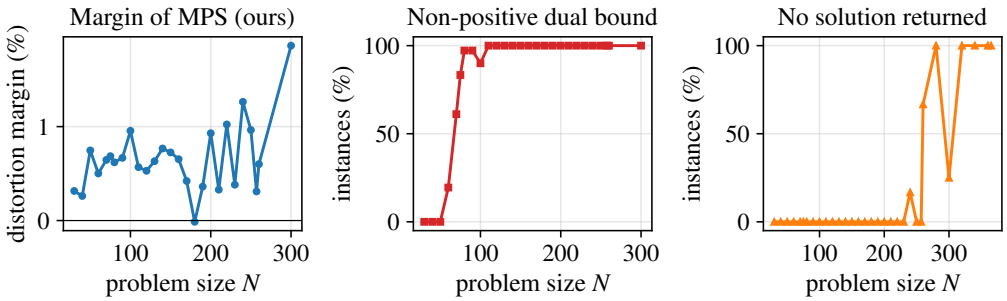


Figure 1: Distortion margin of MPS (ours) over FHG (2025) [39], and the shares of instances on which the bound of FHG is non-positive and on which it returns no solution, against problem size.

The margin over the dual solver is consistent across instances and datasets. At one hundred and fifty classes and above, on every instance on which both return a solution, our solver attains a lower distortion than the dual solver or one on a par with it, differing by less than two tenths of a percent. The mean relative difference is 0.46% on Caltech-256 and 1.27% on Places365.

MPS also matches more classes correctly than the dual solver. Table 2 reports matching accuracy, the quantity of practical interest, aggregated over bands of problem size, since at a single size the estimate rests on few instances once N is large and changes direction between consecutive sizes at almost every step. Our solver is the more accurate of the two in every band, and it stays between roughly three and six times the chance level throughout, with no downward trend in N (Spearman $\rho = +0.00$, $p = 0.91$ across instances). In absolute terms accuracy falls as the problem grows, from above seven percent in the smallest band to below three percent in the largest (Spearman $\rho = -0.24$, $p < 10^{-8}$), in step with the chance level $1/N$, which falls at least as fast. Because our solver optimizes the matching criterion better than any competing method, it also exposes the remaining gap as a property of the criterion itself, since accuracy stays well below its ceiling even at the lowest distortion.

5.5 Solution Availability and Optimality Evidence at Scale

Our solver returns a feasible matching at every size we evaluate, and for the dual solver we read its own diagnostics, the lower bound and the availability of a solution. It proves optimality on none of the six hundred and twelve instances on which it returns a permutation. Since the distortion is non-negative, a lower bound that is itself non-positive excludes nothing, and while at thirty classes the bound remains informative, the fraction of instances carrying a non-positive bound grows with N until, from one hundred and ten classes onward, every bound we record is non-positive. The availability of a solution fails in its turn. Up to two hundred and fifty classes a permutation is returned on all but one instance we ran, beyond roughly two hundred and sixty the solver frequently returns none within twice its budget, and past three hundred and twenty it returns none. Figure 1 shows the three readings together against problem size, with the supporting counts in Section G.4.

5.6 Further Results

Four further studies are reported in the appendix, each with its measurements and its discussion in place, namely the component ablation against exhaustively enumerated global optima (Section F), the evaluation on QAPLIB (Section G.1), the encoder zoo at complete class sets (Section G.5), and a repetition under a substantially longer time limit (Section G.3).

6 Discussion and Limitations

Problem (3) is non-convex, and for such problems no algorithm can guarantee a global maximizer [3]; we establish monotone ascent, and only convergence of the objective sequence and finite termination at a local optimum of the exchange neighbourhood can be proved.

7 Conclusion

This paper has shown that blind vision–language matching can be solved at the scale of several hundred classes without any relaxation, in the sense of losing tightness, and without dual machinery. The Koopmans–Beckmann reduction removed the fourth-order cost tensor that bounds the attainable problem size, the spectral shift exposed why a single trajectory halts short of the optimum and identified breadth of search as the variable that matters, and a closed-form identity for the exchange neighbourhood made that breadth affordable. We proved monotone ascent, convergence and finite termination, and on three datasets the resulting solver outperforms state-of-the-art and benchmark methods in distortion at every problem size and, on average, matches more classes correctly than the dual solver. Over the same range the dual route no longer obtains the guarantee of optimality that motivates it, its lower bound degrading until it excludes nothing, whereas our solver returns a feasible matching at every size and whenever it is stopped.

AI use statement

AI-assisted editing was used for language editing and proofreading of the main paper and supplementary materials. The authors take full responsibility for the final content of this paper.

Reproducibility statement

The solver is specified completely in Section 3 and in Algorithm 1 of Section D; the assumptions behind Lemma 1, Proposition 1, Lemma 2, Lemma 3 and Proposition 2 are stated where the results are, and the proofs are given in Section C. The datasets, the construction of the two kernels, the prompt templates and the renormalization convention are described in Section 5.3, and the two aspects of the comparison most likely to affect the outcome, namely the matched wall-clock budget and its external enforcement, are described in Section 5.1 and quantified in Section G.3. We provide as supplementary material an archive containing the source code of our solver, the dataset and encoder definitions, the scripts that extract the class representations and build the kernels, the evaluation and analysis scripts, and the result files behind every number we report, together with a README that states how each table and figure is reproduced.

Every solver run reported here was executed on one desktop machine, an Intel Core i7-13700K with sixteen cores and twenty-four threads and thirty-two gigabytes of memory. Every timing we report is elapsed wall-clock time, measured on the process from the moment the solver is handed its kernels, and it is wall-clock time that the budget is enforced against; we do not report processor time. The graphics card is used only to extract the class representations, so no stage of the solving is accelerated and the elapsed time of a run reflects work done on the processor alone. Because the comparison is budget-matched, with both solvers of any pair run under identical conditions on the same machine, those figures are directly comparable.

References

[1] David Alvarez-Melis and Tommi S. Jaakkola. Gromov-Wasserstein alignment of word embedding spaces. In *EMNLP*, 2018.

[2] Prabhu Babu and Petre Stoica. Fair principal component analysis (PCA): minorization-maximization algorithms for Fair PCA, Fair Robust PCA and Fair Sparse PCA. *arXiv:2305.05963*, 2023.

[3] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[4] Rainer E. Burkard, Stefan E. Karisch, and Franz Rendl. QAPLIB – a quadratic assignment problem library. *Journal of Global Optimization*, 10(4):391–403, 1997.

[5] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021.

[6] Samir Chowdhury, David Miller, and Tom Needham. Quantized Gromov-Wasserstein. In *ECML PKDD*, 2021.

[7] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *NeurIPS*, 2013.

[8] Donniell E. Fishkind, Sancar Adali, Heather G. Patsolic, Lingyao Meng, Digvijay Singh, Vince Lyzinski, and Carey E. Priebe. Seeded graph matching. *Pattern Recognition*, 87:203–215, 2019.

[9] Steven Gold and Anand Rangarajan. A graduated assignment algorithm for graph matching. *IEEE TPAMI*, 18(4):377–388, 1996.

[10] Edouard Grave, Armand Joulin, and Quentin Berthet. Unsupervised alignment of embeddings with Wasserstein Procrustes. In *AISTATS*, 2019.

[11] Gregory Griffin, Alex Holub, and Pietro Perona. Caltech-256 object category dataset. Technical Report 7694, California Institute of Technology, 2007.

[12] Peter Hahn and Thomas Grant. Lower bounds for the quadratic assignment problem based upon a dual formulation. *Operations Research*, 46(6):912–922, 1998.

[13] Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. Position: The platonic representation hypothesis. In *ICML*, 2024.

[14] Lisa Hutschenreiter, Stefan Haller, Lorenz Feineis, Carsten Rother, Dagmar Kainmüller, and Bogdan Savchynskyy. Fusion moves for graph matching. In *ICCV*, 2021.

[15] Tjalling C. Koopmans and Martin Beckmann. Assignment problems and the location of economic activities. *Econometrica*, 25:53–76, 1957.

[16] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *ICML*, 2019.

[17] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

[18] Harold W. Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 1955.

[19] Eugene L. Lawler. The quadratic assignment problem. *Management Science*, 9(4):586–599, 1963.

[20] Marc Lelarge. Chaining 2-FWL GNNs for combinatorial graph alignment. *arXiv:2510.03086*, 2025.

[21] Jiajin Li, Jianheng Tang, Lemin Kong, Huikang Liu, Jia Li, Anthony Man-Cho So, and Jose Blanchet. A convergent single-loop algorithm for relaxation of Gromov-Wasserstein in graph data. In *ICLR*, 2023.

[22] Ling Liang and Lei Yang. A provably convergent and practical algorithm for Gromov–Wasserstein optimal transport. *arXiv:2605.04175*, 2026.

[23] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A ConvNet for the 2020s. In *CVPR*, 2022.

[24] Mayug Maniparambil, Raiymbek Akshulakov, Yasser Abdelaziz Dahou Djilali, Mohamed El Amine Seddik, Sanath Narayan, Karttikeya Mangalam, and Noel E. O'Connor. Do vision and language encoders represent the world similarly? In *CVPR*, 2024.

[25] Facundo Mémoli. Gromov-Wasserstein distances and the metric approach to object matching. *Foundations of Computational Mathematics*, 11:417–487, 2011.

[26] Peter Merz and Bernd Freisleben. Fitness landscape analysis and memetic algorithms for the quadratic assignment problem. *IEEE Transactions on Evolutionary Computation*, 4(4):337–352, 2000.

[27] Pablo Moscato. On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms. Technical report, Caltech Concurrent Computation Program, C3P Report 826, 1989.

[28] Luca Moschella, Valentino Maiorca, Marco Fumero, Antonio Norelli, Francesco Locatello, and Emanuele Rodolà. Relative representations enable zero-shot latent space communication. In *ICLR*, 2023.

[29] Antonio Norelli, Marco Fumero, Valentino Maiorca, Luca Moschella, Emanuele Rodolà, and Francesco Locatello. ASIF: Coupled data turns unimodal models to multimodal without training. In *NeurIPS*, 2023.

[30] Mohammad Mahdi Omati, Prabhu Babu, Petre Stoica, and Arash Amini. A max-min approach to the worst-case class separation problem. *Transactions on Machine Learning Research*, 2025. URL `https://openreview.net/forum?id=EEmwBd4tfZ`.

[31] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.

[32] Gabriel Peyré, Marco Cuturi, and Justin Solomon. Gromov-Wasserstein averaging of kernel and distance matrices. In *ICML*, 2016.

[33] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

[34] Anand Rangarajan, Alan L. Yuille, Steven Gold, and Eric Mjolsness. A convergence proof for the softassign quadratic assignment algorithm. In *NeurIPS*, 1996.

[35] Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using siamese BERT-networks. In *EMNLP-IJCNLP*, 2019.

[36] Gabriel Rioux, Ziv Goldfeld, and Kengo Kato. Entropic Gromov-Wasserstein distances: Stability and algorithms. *Journal of Machine Learning Research*, 25, 2024.

[37] Sartaj Sahni and Teofilo Gonzalez. P-complete approximation problems. *Journal of the ACM*, 23(3):555–565, 1976.

[38] Meyer Scetbon, Gabriel Peyré, and Marco Cuturi. Linear-time Gromov Wasserstein distances using low rank couplings and costs. In *ICML*, 2022.

[39] Dominik Schnaus, Nikita Araslanov, and Daniel Cremers. It's a (Blind) Match! Towards vision-language correspondence without parallel data. In *CVPR*, 2025.

[40] Liangliang Shi, Haoran Zhang, Shuheng Shen, Changhua Meng, Weiqiang Wang, and Junchi Yan. BiQAP: Neural bi-level optimization-based framework for solving quadratic assignment problems. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, 2025. doi: 10.1145/3711896.3736859.

[41] Ying Sun, Prabhu Babu, and Daniel P. Palomar. Majorization-minimization algorithms in signal processing, communications, and machine learning. *IEEE Transactions on Signal Processing*, 65(3):794–816, 2017.

[42] Éric Taillard. Robust taboo search for the quadratic assignment problem. *Parallel Computing*, 17(4-5):443–455, 1991.

[43] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *ICML*, 2021.

[44] Sanketh Vedula, Valentino Maiorca, Lorenzo Basile, Francesco Locatello, and Alex Bronstein. Scalable unsupervised alignment of general metric and non-metric structures. *arXiv:2406.13507; ICML AI4Science Workshop*, 2024.

[45] Cédric Vincent-Cuaz, Rémi Flamary, Marco Corneli, Titouan Vayer, and Nicolas Courty. Semi-relaxed Gromov-Wasserstein divergence and applications on graphs. In *ICLR*, 2022.

[46] Joshua T. Vogelstein, John M. Conroy, Vince Lyzinski, Louis J. Podrazik, Steven G. Kratzer, Eric T. Harley, Donniell E. Fishkind, R. Jacob Vogelstein, and Carey E. Priebe. Fast approximate quadratic programming for graph matching. *PLOS One*, 10(4), 2015.

[47] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE TPAMI*, 40(6):1452–1464, 2018.

APPENDIX

This appendix is organized as follows. Section A reviews the minorization–maximization framework on which our development rests, and Section B places the work within the literature at greater length than the introduction allows. Section C gives complete proofs of every claim stated in the main text, in the order in which the claims appear there, and Section D states the complete procedure. Section E describes the datasets and the encoders, Section F isolates the contribution of each component of the solver against exhaustively enumerated global optima and measures what the choice of initialization costs, and Section G reports the remaining measurements.

A AN OVERVIEW OF THE MM FRAMEWORK

Consider the constrained optimization problem

$$\max_{\mathbf{x}\in\chi} f(\mathbf{x}), \tag{S1}$$

where $\mathbf{x}$ denotes the decision variable, $f(\mathbf{x})$ is the objective to be maximized and χ represents the feasible region. Building on minorization–maximization principles, one tackles (S1) by introducing, at each iteration t, a surrogate function $g(\mathbf{x} \mid \mathbf{x}^t)$ which underestimates $f(\mathbf{x})$ but matches it exactly at the current point $\mathbf{x}^t$; the next iterate is then found by solving $\mathbf{x}^{t+1} \in \arg\max_{\mathbf{x}\in\chi} g(\mathbf{x} \mid \mathbf{x}^t)$, and these two operations, surrogate construction and maximization, are repeated until convergence. For $g(\mathbf{x} \mid \mathbf{x}^t)$ to qualify as a valid minorizer it must satisfy

$$g\big(\mathbf{x} \mid \mathbf{x}^t\big) \le f(\mathbf{x}) \quad \forall\, \mathbf{x} \in \chi, \qquad g\big(\mathbf{x}^t \mid \mathbf{x}^t\big) = f\big(\mathbf{x}^t\big), \tag{S2}$$

and as a result of these properties each iteration yields

$$f\big(\mathbf{x}^{t+1}\big) \;\ge\; g\big(\mathbf{x}^{t+1} \mid \mathbf{x}^t\big) \;\ge\; g\big(\mathbf{x}^t \mid \mathbf{x}^t\big) \;=\; f\big(\mathbf{x}^t\big), \tag{S3}$$

which shows that the objective value never decreases, ensuring that the sequence $f(\mathbf{x}^t)$ converges whenever it is bounded above. The first inequality holds because g minorizes f everywhere on χ, the second because $\mathbf{x}^{t+1}$ maximizes the surrogate over χ while $\mathbf{x}^t$ is merely feasible for it, and the final equality is the tangency condition in (S2). For a fuller account, please refer to [41].

Two features of this framework govern how it is used in the main text. First, MM is a design principle, and what one obtains from it depends entirely on the surrogate one is able to construct, so the whole difficulty of applying it to a new problem lies in finding a minorizer at once tight enough for the iterations to make progress and simple enough to be maximized exactly. Second, the guarantee (S3) concerns the objective sequence alone, and further argument is required before anything may be concluded about the sequence of iterates itself. Both points are taken up in Section 3.1, where a minorizer is constructed for the matching objective, and in Section 4.3, where the behaviour of the iterates is established.

B RELATED WORK

The observation that independently trained models converge toward related representational geometry has been made from several directions. The platonic representation hypothesis states it in general form, and several methods exploit it to connect unimodal encoders through a sparse set of anchors, which the fully unsupervised setting does not provide. LocalCKA studies the same convergence through centered kernel alignment, and on symmetric kernels its matching step coincides with the Frank–Wolfe scheme discussed below and inherits its single-trajectory limitation. The fully unsupervised setting was formalized as a quadratic assignment over pairwise distances between class representations, and its evaluation has so far been confined to at most one hundred classes by the datasets employed. A related use of the Gromov–Wasserstein distance for correspondence is unsupervised word translation [1].

Lower bounds on the optimum of the quadratic assignment problem have long been constructed by Lagrangian ascent, of which the Hahn–Grant method is the most directly relevant here. Fusion moves combine search with bounding on the Lawler form of the problem, whose cost tensor grows with the fourth power of the number of classes once every pair of assignments carries a cost, as it

does here [14]. The price of the dual route on blind matching is $O(N^5)$ operations per iteration and $O(N^3)$ memory, paid for a guarantee of optimality that holds only while the bound remains informative, and for an objective that is non-negative by construction the bound carries no information once it turns non-positive. We measure the scale at which that happens, and that the guarantee is already unavailable at a hundred classes does not appear to have been reported.

Primal approaches include the Frank–Wolfe scheme FAQ, pairwise-exchange local search, whose incremental neighbourhood evaluation goes back to robust tabu search, memetic algorithms, long among the strongest heuristics for this problem, and graduated assignment with its softassign iteration [9, 34]. What FAQ, pairwise exchange and graduated assignment share is that a single trajectory is followed to the first point at which it ceases to improve, and the vertices at which such a trajectory can come to rest are many and differ considerably in objective value, so the quality of the result is settled by where the trajectory happened to begin. Our solver follows the memetic template, and what we add to it is the convexified minorization–maximization analysis of Section 3.1, which explains why single-trajectory schemes come to rest short of the optimum on this objective, together with a closed-form evaluation of the complete exchange neighbourhood in a single matrix product, which makes population-based search affordable once N reaches the hundreds. Scalable solvers for the Gromov–Wasserstein problem have appeared steadily since the entropic Gromov–Wasserstein iteration, including entropic regularization with convergence guarantees, the single-loop scheme BAPG, low-rank couplings, quantized formulations, semi-relaxed divergences and inexact projected-gradient schemes [36, 38, 6, 45, 22]. The entropic, low-rank, quantized, semi-relaxed and projected-gradient solvers return couplings that are not bijections between the two class sets, which is the object this problem requires. Learning-based solvers require a training distribution of instances [40, 44, 20]. They address the inductive regime, whereas each matching here is a single instance with no distribution to learn from. Wasserstein–Procrustes learns a mapping between the two embedding spaces [10]. Methods of this kind are not well posed here, since with a few hundred classes in several hundred dimensions a linear map can align almost any pair of point sets.

C Proofs

C.1 Reduction to the Koopmans–Beckmann form

We restate and prove Lemma 1. Expanding the square in (2) separates the objective into three summations,

$$F(\mathbf{P}) = \underbrace{\sum_{i,j,k,l} X_{ik}^2 P_{ij} P_{kl}}_{T_1} - 2 \underbrace{\sum_{i,j,k,l} X_{ik} Y_{jl} P_{ij} P_{kl}}_{T_2} + \underbrace{\sum_{i,j,k,l} Y_{jl}^2 P_{ij} P_{kl}}_{T_3}, \tag{S4}$$

which we treat in turn. In T_1 the coefficient X_{ik}^2 does not involve the indices j and l, so those summations may be evaluated independently,

$$T_1 = \sum_{i,k} X_{ik}^2 \sum_{j,l} P_{ij} P_{kl} = \sum_{i,k} X_{ik}^2 \Big(\sum_j P_{ij} \Big) \Big(\sum_l P_{kl} \Big) = \sum_{i,k} X_{ik}^2 = \|\mathbf{X}\|_F^2, \tag{S5}$$

where the third equality uses that every row of a permutation matrix sums to unity. The term T_3 is symmetric to T_1, with the roles of the row and column sums interchanged, since the coefficient Y_{jl}^2 does not involve i and k, and $\sum_i P_{ij} = \sum_k P_{kl} = 1$ because every column of a permutation matrix likewise sums to unity, whence $T_3 = \|\mathbf{Y}\|_F^2$. Neither term depends on $\mathbf{P}$.

Only T_2 retains a dependence on the assignment. Reordering its summations so that the innermost involves l alone,

$$T_2 = \sum_{i,k} X_{ik} \sum_j P_{ij} \sum_l Y_{jl} P_{kl}, \tag{S6}$$

the innermost sum $\sum_l Y_{jl} P_{kl}$ is the (j,k) entry of $\mathbf{Y}\mathbf{P}^\top$, and the sum over j that follows contracts P_{ij} with it to give the (i,k) entry of $\mathbf{P}\mathbf{Y}\mathbf{P}^\top$. Hence

$$T_2 = \sum_{i,k} X_{ik} \big[\mathbf{P}\mathbf{Y}\mathbf{P}^\top\big]_{ik} = \big\langle \mathbf{X},\, \mathbf{P}\mathbf{Y}\mathbf{P}^\top \big\rangle_F = \operatorname{tr}\big(\mathbf{X}^\top \mathbf{P}\mathbf{Y}\mathbf{P}^\top\big) = \operatorname{tr}\big(\mathbf{P}^\top \mathbf{X}\mathbf{P}\mathbf{Y}\big), \tag{S7}$$

the last step following from the cyclic invariance of the trace together with the symmetry of $\mathbf{X}$. Substituting the three terms into (S4) gives the stated identity, and since $\|\mathbf{X}\|_F^2$ and $\|\mathbf{Y}\|_F^2$ are constants over $\mathcal{P}_N$, the minimizer of F is the maximizer of the trace. ■

C.2 Invariance of the discrete optima under the spectral shift

We restate and prove Proposition 1. Let $\tilde{\mathbf{X}} = \mathbf{X} - \alpha\mathbf{I}$ and $\tilde{\mathbf{Y}} = \mathbf{Y} - \beta\mathbf{I}$ with $\alpha = \lambda_{\min}(\mathbf{X})$ and $\beta = \lambda_{\min}(\mathbf{Y})$. For any $\mathbf{P} \in \mathcal{P}_N$,

$$\begin{aligned}\operatorname{tr}\big(\mathbf{P}^\top \tilde{\mathbf{X}} \mathbf{P} \tilde{\mathbf{Y}}\big) &= \operatorname{tr}\big(\mathbf{P}^\top (\mathbf{X} - \alpha\mathbf{I}) \mathbf{P} (\mathbf{Y} - \beta\mathbf{I})\big) \\ &= \operatorname{tr}\big(\mathbf{P}^\top \mathbf{X} \mathbf{P} \mathbf{Y}\big) - \beta \operatorname{tr}\big(\mathbf{P}^\top \mathbf{X} \mathbf{P}\big) - \alpha \operatorname{tr}\big(\mathbf{P}^\top \mathbf{P} \mathbf{Y}\big) + \alpha\beta \operatorname{tr}\big(\mathbf{P}^\top \mathbf{P}\big). \end{aligned} \tag{S8}$$

Because $\mathbf{P}$ is a permutation matrix we have $\mathbf{P}^\top\mathbf{P} = \mathbf{P}\mathbf{P}^\top = \mathbf{I}$, and hence by cyclicity of the trace $\operatorname{tr}(\mathbf{P}^\top \mathbf{X} \mathbf{P}) = \operatorname{tr}(\mathbf{X}\mathbf{P}\mathbf{P}^\top) = \operatorname{tr}(\mathbf{X})$, similarly $\operatorname{tr}(\mathbf{P}^\top \mathbf{P} \mathbf{Y}) = \operatorname{tr}(\mathbf{Y})$, and $\operatorname{tr}(\mathbf{P}^\top \mathbf{P}) = N$. Substituting gives

$$\operatorname{tr}\big(\mathbf{P}^\top \tilde{\mathbf{X}} \mathbf{P} \tilde{\mathbf{Y}}\big) = \operatorname{tr}\big(\mathbf{P}^\top \mathbf{X} \mathbf{P} \mathbf{Y}\big) - \beta \operatorname{tr}(\mathbf{X}) - \alpha \operatorname{tr}(\mathbf{Y}) + \alpha\beta N, \tag{S9}$$

in which the last three terms do not depend on $\mathbf{P}$. The two objectives therefore differ by an additive constant over $\mathcal{P}_N$ and are maximized by exactly the same permutations. ■

C.3 Convexity of the shifted objective

We restate and prove Lemma 2. Writing $\mathbf{p} = \operatorname{vec}(\mathbf{P})$, the identity $\operatorname{vec}(\mathbf{APB}) = (\mathbf{B}^\top \otimes \mathbf{A})\mathbf{p}$ gives

$$\operatorname{tr}\big(\mathbf{P}^\top \tilde{\mathbf{X}} \mathbf{P} \tilde{\mathbf{Y}}\big) = \mathbf{p}^\top \big(\tilde{\mathbf{Y}} \otimes \tilde{\mathbf{X}}\big) \mathbf{p}, \tag{S10}$$

using the symmetry of $\tilde{\mathbf{Y}}$. The Hessian with respect to $\mathbf{p}$ is $2(\tilde{\mathbf{Y}} \otimes \tilde{\mathbf{X}})$. The eigenvalues of a Kronecker product are the pairwise products of the eigenvalues of its factors, and both factors are positive semidefinite by construction of the shift, so every eigenvalue of the Hessian is non-negative and the shifted objective is convex on $\mathbb{R}^{N \times N}$. ■

C.4 The exchange-neighbourhood identity

We restate and prove Lemma 3, giving the corrections that the statement in the main text collects into $\mathbf{K}^{(1)}$, $\mathbf{K}^{(2)}$ and $\mathbf{C}$. Let $\mathbf{B} = \mathbf{Y}[\pi][:, \pi]$, let $\mathbf{S} = \mathbf{XB}$ and $r_i = \sum_j X_{ij} B_{ij}$, and write $d_i^X = X_{ii}$ and $d_i^B = B_{ii}$ for the diagonals. Exchanging the assignments at positions a and b permutes rows and columns a and b of $\mathbf{B}$, and for symmetric $\mathbf{X}$ and $\mathbf{B}$ the induced change in $\sum_{i,k} X_{ik} B_{ik}$ is

$$\Delta_{ab} = 2 \sum_{k \notin \{a,b\}} \big(X_{ak} - X_{bk}\big)\big(B_{bk} - B_{ak}\big) + c_{ab}, \tag{S11}$$

where c_{ab} collects the terms with $k \in \{a, b\}$. The unrestricted sum over all k is expressible through $\mathbf{S}$ and $\mathbf{r}$ as $S_{ab} + S_{ba} - r_a - r_b$, so that in matrix form

$$\mathbf{T} = \mathbf{S} + \mathbf{S}^\top - \mathbf{r}\mathbf{1}^\top - \mathbf{1}\mathbf{r}^\top. \tag{S12}$$

Subtracting the two excluded indices gives the corrections

$$(\mathbf{K}^{(1)})_{ab} = \big(d_a^X - X_{ab}\big)\big(B_{ab} - d_a^B\big), \qquad (\mathbf{K}^{(2)})_{ab} = \big(X_{ab} - d_b^X\big)\big(d_b^B - B_{ab}\big), \tag{S13}$$

and the diagonal block contributes

$$C_{ab} = \big(d_a^X - d_b^X\big)\big(d_b^B - d_a^B\big), \tag{S14}$$

whence $\mathbf{\Delta} = 2(\mathbf{T} - \mathbf{K}^{(1)} - \mathbf{K}^{(2)}) + \mathbf{C}$ as stated. Each correction is an entrywise product of two rank-one differences, dense in general, and it is inexpensive because it is read off the diagonals and the (a, b) entries with no further matrix product. ■

The dominant cost is the single product $\mathbf{XB}$, so the whole neighbourhood is obtained in $O(N^3)$ and the amortized cost per candidate move is $O(N)$. We verified the identity against direct recomputation of the objective for randomly generated instances up to $N = 200$, agreeing to machine precision; that verification is part of the released material.

C.5 MONOTONICITY, BOUNDEDNESS AND CONVERGENCE

We prove Proposition 2, treating its four assertions in turn, and then the statement concerning the incumbent.

(a) Monotonicity. Consider first the exchange search. By construction an exchange is applied only when its delta, computed exactly through Lemma 3, is strictly positive, so the objective strictly increases at every accepted move. Consider next the linearization iteration (6). By Lemma 2 the shifted objective $\tilde{f}(\mathbf{P}) = \operatorname{tr}(\mathbf{P}^\top \tilde{\mathbf{X}} \mathbf{P} \tilde{\mathbf{Y}})$ is convex, so its first-order expansion at $\mathbf{P}^t$ is a global underestimator,

$$\tilde{f}(\mathbf{P}) \;\geq\; \tilde{f}\big(\mathbf{P}^t\big) + \operatorname{tr}\big((\mathbf{G}^t)^\top (\mathbf{P} - \mathbf{P}^t)\big) \qquad \text{for all } \mathbf{P}, \tag{S15}$$

with $\mathbf{G}^t = 2\tilde{\mathbf{X}}\mathbf{P}^t\tilde{\mathbf{Y}}$. Evaluating (S15) at the maximizer $\mathbf{P}^{t+1}$ of (6), and using $\operatorname{tr}((\mathbf{G}^t)^\top \mathbf{P}^{t+1}) \geq \operatorname{tr}((\mathbf{G}^t)^\top \mathbf{P}^t)$, which holds because $\mathbf{P}^t$ is itself feasible for the linear assignment problem that $\mathbf{P}^{t+1}$ solves, gives $\tilde{f}(\mathbf{P}^{t+1}) \geq \tilde{f}(\mathbf{P}^t)$. By Proposition 1 the shift alters the objective by an additive constant only, so monotonicity in $\tilde{f}$ is equivalent to monotonicity in f. This is the instance, for our minorizer, of the general chain (S3).

(b) Boundedness. Monotonicity alone does not imply convergence, and we therefore exhibit bounds on the objective. Writing it as a Frobenius inner product and applying the Cauchy–Schwarz inequality,

$$f(\mathbf{P}) = \big\langle \mathbf{X},\, \mathbf{P}\mathbf{Y}\mathbf{P}^\top \big\rangle_F \;\leq\; \|\mathbf{X}\|_F \,\big\|\mathbf{P}\mathbf{Y}\mathbf{P}^\top\big\|_F \;=\; \|\mathbf{X}\|_F \,\|\mathbf{Y}\|_F, \tag{S16}$$

where the final equality holds because the Frobenius norm is invariant under multiplication by orthogonal matrices and $\mathbf{P}^\top\mathbf{P} = \mathbf{P}\mathbf{P}^\top = \mathbf{I}$ for every permutation matrix. A lower bound is immediate from the entrywise non-negativity of the two distance kernels, which gives $f(\mathbf{P}) = \sum_{i,k} X_{ik}[\mathbf{P}\mathbf{Y}\mathbf{P}^\top]_{ik} \geq 0$. Both bounds are independent of $\mathbf{P}$, hence of t, so the objective sequence is confined to a bounded interval determined by the data alone.

(c) Convergence of the objective sequence. By (a) the sequence $\{f(\mathbf{P}^t)\}$ is non-decreasing and by (b) it is bounded above. By the monotone convergence theorem it therefore converges to a finite limit $f^\infty \leq \|\mathbf{X}\|_F \|\mathbf{Y}\|_F$. In the present setting a stronger statement holds, because every iterate lies in $\mathcal{P}_N$, which is a finite set, the objective takes at most $N!$ distinct values, and a non-decreasing sequence drawn from a finite set is eventually constant. The limit f^∞ is thus attained after finitely many iterations.

(d) Convergence of the iterate sequence. It remains to establish that the iterates themselves settle, which does not follow from (c) alone, since an objective sequence may become constant while the arguments continue to move. For the exchange search, strict increase at every accepted move means no permutation can be revisited; as $\mathcal{P}_N$ is finite, the search must halt after finitely many rounds, and by the acceptance rule it halts only when no exchange carries a positive delta, that is, at a local optimum of the exchange neighbourhood. For the linearization iteration the argument needs one more step, because a non-decreasing objective on a finite set forces the objective *value* to become constant but not the iterate to stop, since the sequence could in principle cycle among distinct permutations sharing that value. The tie-breaking convention introduced with (6) removes this possibility. By (c) there is a t_0 beyond which $f(\mathbf{P}^t)$ is constant, and (S15) evaluated at $\mathbf{P}^{t+1}$ then forces $\operatorname{tr}((\mathbf{G}^t)^\top \mathbf{P}^{t+1}) = \operatorname{tr}((\mathbf{G}^t)^\top \mathbf{P}^t)$ for every $t \geq t_0$, so that $\mathbf{P}^t$ is itself a maximizer of the linearized problem it poses and the convention returns $\mathbf{P}^{t+1} = \mathbf{P}^t$. The sequence is therefore constant from t_0 onward, at a point $\mathbf{P}^\infty$ with $\mathbf{P}^\infty \in \arg\max_{\mathbf{P}\in\mathcal{P}_N} \operatorname{tr}((\mathbf{G}^\infty)^\top \mathbf{P})$, where $\mathbf{G}^\infty = 2\tilde{\mathbf{X}}\mathbf{P}^\infty\tilde{\mathbf{Y}}$. Such a point is precisely one that is optimal against its own linearization, which by (S15) is the first-order stationarity condition for (3) restricted to $\mathcal{P}_N$. The two iterations therefore terminate finitely, the exchange search at a local optimum of the exchange neighbourhood and the linearization iteration at a maximizer of the linear problem it poses, and neither condition implies the other.

The incumbent. Finally, the incumbent of MPS is the population maximum, and a child is admitted only when it strictly exceeds the weakest member. The population maximum is therefore non-decreasing over the course of the run, and because a feasible permutation exists from the first iteration onward, the procedure may be interrupted at any moment and still return one. ■

D THE COMPLETE ALGORITHM

For the convenience of the reader, the components developed in Section 3 are assembled in Algorithm 1, with the local refinement stated separately as Algorithm 2. The procedure first constructs a population $\mathcal{Q}$ of M distinct local optima by running the local search from random permutations, and thereafter repeatedly selects two members at random, recombines them through (7), refines the child, and admits it under the replacement rule of Section 3.3. The incumbent is the population maximum and is available at every instant, so that the procedure is anytime in the strict sense, in that it consumes whatever wall-clock budget it is granted and may be stopped at any time with a feasible permutation available. We fix $M = 10$ throughout and use no other tuned parameter in any experiment reported in this paper.

The local step of the solver whose results we report is the exchange search of Algorithm 2. The linearization iteration (6) serves the analysis of Section 3.1 and is measured as a baseline in Section F, and the released implementation also offers it as an optional step. Since the spectral shift alters the objective by an additive constant (Proposition 1), it leaves every exchange delta unchanged, so the exchange search operates on the original kernels.

Algorithm 1 Memetic Primal Search (MPS)

Require: kernels $\mathbf{X}, \mathbf{Y}$; budget T; population size M
$\mathcal{Q} \leftarrow \emptyset$; $t \leftarrow 0$ ▷ t: elapsed time, read at each test
while $|\mathcal{Q}| < M$ **and** $t < T/2$ **do**
  $\pi \leftarrow \textsc{Refine}(\text{random } \pi)$
  if $\pi \notin \mathcal{Q}$ **then** $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{\pi\}$
  end if
end while
sort $\mathcal{Q}$ by f descending
while $t < T$ **and** $|\mathcal{Q}| \geq 2$ **do**
  draw $\pi_a \neq \pi_b$ uniformly from $\mathcal{Q}$
  form π_c by (7)
  $\pi_c \leftarrow \textsc{Refine}(\pi_c)$
  if $f(\pi_c) > \min_{\pi \in \mathcal{Q}} f(\pi)$ **and** $\pi_c \notin \mathcal{Q}$ **then**
    replace the weakest by π_c; re-sort $\mathcal{Q}$
  end if
end while
Output: $\arg\max_{\pi \in \mathcal{Q}} f(\pi)$

Algorithm 2 Local search $\textsc{Refine}(\pi)$

Require: permutation π
repeat ▷ best-improvement exchange search
  $\mathbf{B} \leftarrow \mathbf{Y}[\pi][:, \pi]$
  $\mathbf{\Delta} \leftarrow$ one-product form of Lemma 3
  $(a, b) \leftarrow \arg\max_{i<j} \Delta_{ij}$
  if $\Delta_{ab} \leq 0$ **then break**
  end if
  exchange $\pi(a) \leftrightarrow \pi(b)$
until no improving exchange
Output: π

E DATASETS AND IMPLEMENTATION DETAILS

The datasets of the main comparison are CIFAR-100, Caltech-256 and Places365, and the ablation at ten classes uses CIFAR-10, all taken from their official sources. Places365 is used at its validation split and reduced resolution, since a per-class mean embedding is all this task requires, and Caltech-256 keeps its `clutter` category, so that its class count matches the published size of the dataset. Class names are normalised to natural language before they are embedded, because the raw directory labels some loaders expose would otherwise make the language side meaningless.

The vision side spans thirty-two encoders from the CLIP, ConvNeXt, DeiT, DINO and DINOv2 families [33, 23, 43, 5, 31]. The language side spans twenty-seven encoders from the Sentence-Transformers and CLIP text families [35]. Both lists are reproduced in the released material together with the identifier under which each checkpoint is obtained. The comparison of Section 5 uses six of the vision encoders and two of the language encoders, and the study of Section G.5 uses all thirty-two vision encoders against one language encoder per dataset. Budgets are one hundred and eighty seconds per solver below one hundred and fifty classes and six hundred seconds above, with the external termination at twice those values described in Section 5.1.

Table S1: Component ablation at $N = 10$, against exhaustively enumerated global optima. Shaded row is ours. **Bold** = best, underlined = second best.

variant	accuracy (%)	global optimum (%)
exchange search, single start	20.5 ± 21.1	20
multi-start, $R = 5$	$\mathbf{46.0 \pm 20.6}$	55
multi-start, $R = 20$	$\underline{33.0 \pm 7.3}$	**100**
multi-start, $R = 50$	$\underline{33.0 \pm 7.3}$	**100**
linearization iteration, $R = 100$	22.5 ± 14.8	0
linearization, progressive shift, $R = 20$	32.0 ± 8.9	$\underline{95}$
MPS (ours)	$\underline{33.0 \pm 7.3}$	**100**

F Methodological Ablation Study

Before comparing MPS against the published solvers on blind matching, we establish which of its components are responsible for its behaviour, and at which problem sizes each of them matters. This is a prerequisite for interpreting the comparison of Section 5, since a solver whose advantage came from its restarts alone would not support the argument of Section 3.1. The two tables below, like the anytime measurement of Section G.2, pose problem (3) on the inner-product kernels $X_{ik} = \mathbf{u}_i^\top \mathbf{u}_k$ and $Y_{jl} = \mathbf{v}_j^\top \mathbf{v}_l$ of the class means, a second pair of symmetric kernels for the same Koopmans–Beckmann problem, and each of their instances forms the vision class means from a random half of every class's images, drawn afresh for each instance.

We first isolate the contribution of each component at a scale where the ground truth of the optimization is available. At $N = 10$ the global optimum of (3) can be obtained by exhaustive enumeration, so that the fraction of instances on which a variant provably attains it is exactly measurable. Table S1 reports this over twenty such draws of the ten CIFAR-10 classes, matching CLIP ViT-L/14 against all-mpnet-base-v2; its *global* column is that exactly measured fraction, and *accuracy* is the proportion of classes assigned correctly. The variants are the exchange search from a single start; multi-start exchange search with R restarts; the linearization iteration (6) restarted one hundred times; a variant of it in which the spectral shift is reduced progressively; and MPS itself.

As we can see, the pattern confirms the analysis of Section 3.1. A single trajectory of the exchange search reaches the optimum on one instance in five, and the linearization iteration, even restarted one hundred times, reaches it on none, as predicted for a scheme that halts at whichever vertex is optimal against its own linearization. Breadth repairs this, since twenty independent restarts of the exchange search already attain the global optimum on every instance, and MPS attains it likewise. At this scale, therefore, restarts suffice and recombination has nothing left to add.

Applied to a permutation that the exchange search has already brought to a local optimum, the linearization iteration returned that same permutation in every trial, over nine problem sizes from ten classes to three hundred and sixty-five. Every local optimum of the exchange neighbourhood we produced was thus already optimal against its own linearization, so the two stationarity conditions of Proposition 2, distinct in general, coincided on every instance we examined, and the solver we report accordingly uses the exchange search alone. At ten classes the accuracy of every variant varies widely from one draw to the next, with standard deviations of seven to twenty-one points, and accuracy is not the quantity this experiment is designed to measure, which is the global column.

The situation reverses once the problem grows. Table S2 repeats the comparison at $N = 100$, over four such draws of the full class set of CIFAR-100, matching DINOv2 ViT-L/14 against all-mpnet-base-v2 under a matched budget. Enumeration is no longer possible at this size, so the objective (3) must itself serve as the criterion, and larger values are better. The draws differ in difficulty by some eight units of that objective, while within any one of them the variants are separated by less than a single unit, so an average across draws would mostly describe the draws, and each is therefore reported on its own. Multi-start local search, which was sufficient at $N = 10$, and iterated local search, which perturbs the incumbent and applies the exchange search again, are both outperformed by MPS on every draw, and the margin over the strongest of them is consistent from one draw to the next. Taken together the two tables locate the contribution exactly, in that restarts are what matter

Table S2: Component ablation at $N = 100$ under a matched budget (↑). Shaded row is ours. **Bold** = best, underlined = second best.

	draw			
variant	1	2	3	4
multi-start exchange search	980.941	978.528	979.732	986.049
iterated local search	981.224	978.808	979.707	985.940
iterated local search with linearization steps	980.650	978.158	979.785	985.930
MPS (ours)	**981.670**	**979.164**	**980.631**	**986.465**

Table S3: Sensitivity to the initialization at $N \geq 150$. Shaded row is ours. **Bold** = best, underlined = second best.

initialization	distinct members ↑	diversity ↑	penalty (%) ↓	worst case (%) ↓
spectral ordering	1.0	0.000	1.23	6.87
row-sum ordering	1.0	0.000	1.45	8.50
spectral, perturbed	9.9	0.850	0.11	0.63
row-sum, perturbed	9.8	0.866	0.10	0.44
uniform random (ours)	**10.0**	**0.963**	**0.06**	**0.30**

while the search space can still be covered by chance, and agreement-preserving recombination is what matters once it cannot, which is the regime this paper concerns.

F.1 Sensitivity to the initialization

MPS begins from uniformly random permutations, and Section 4.1 argues that the choice is deliberate, since recombination re-searches only the positions on which two parents disagree and a population descended from one structured point has almost nothing left to disagree about. Table S3 measures that argument. Five initializations were compared over twenty-one instances drawn from Caltech-256 and Places365 at $N = 150$ and above, each identical in local search, recombination, budget and seed and differing only in where the ten members of the population begin. Two of them seed every member from a structured ordering, the leading eigenvector of the two kernels in one case and the row sums in the other; two perturb those orderings, member i by i transpositions, which is the form a seeded population would realistically take; and the last is ours. Once each starting point has been refined by the local search, *distinct members* counts the different local optima in the population and *diversity* is the mean pairwise disagreement between the members, as a fraction of N. The penalty column reports the distortion above the best value any variant reached on that same instance, measured instance by instance so that the rows are comparable, and the worst case is its maximum over instances.

The four columns read from left to right as the mechanism they describe. A pure structured start leaves one distinct member out of ten and a diversity of zero, which is to say that every member descends to the same local optimum and the recombination loop never executes, since it requires two parents that differ. What that costs is a penalty above one percent on average and, on a single instance, as much as 8.50%, so the tail is a good deal worse than the mean and the mean alone would understate what a structured start risks.

Perturbing those same orderings recovers almost all of it, and that is what locates the finding. The difficulty lies in a population that shares one origin and so has nothing to recombine, and once the members are made independent of one another the penalty falls by an order of magnitude. Uniform random is the best of the five on every column measured here, and it is the only one that supplies that independence with no parameter to choose, which is why MPS uses it.

Table S4: MPS on 24 QAPLIB instances, against published best known solutions (BKS). **Bold** marks the instances on which the best known value is attained.

instance	n	MPS	BKS	gap (%)	instance	n	MPS	BKS	gap (%)
chr12a	12	9552	9552	**0.000**	nug30	30	6128	6124	0.065
had12	12	1652	1652	**0.000**	tai30a	30	1835972	1818146	0.980
nug12	12	578	578	**0.000**	esc32a	32	130	130	**0.000**
rou12	12	235528	235528	**0.000**	lipa40a	40	31538	31538	**0.000**
tai12a	12	224416	224416	**0.000**	tai40a	40	3189818	3139370	1.607
esc16a	16	68	68	**0.000**	sko42	42	15812	15812	**0.000**
chr20a	20	2222	2192	1.369	wil50	50	48832	48816	0.033
had20	20	6922	6922	**0.000**	tai60a	60	7389374	7205962	2.545
lipa20a	20	3683	3683	**0.000**					
nug20	20	2570	2570	**0.000**	*mean gap*				0.462
rou20	20	725582	725522	0.008	*median gap*				0.000
tai20a	20	703482	703482	**0.000**	*max gap*				2.545
chr25a	25	3884	3796	2.318					
bur26a	26	5455308	5426670	0.528					
kra30a	30	90360	88900	1.642					
lipa30a	30	13178	13178	**0.000**					

Table S5: Anytime behaviour of MPS at $N = 100$. **Bold** = best.

budget (s)	distortion$/N^2$ ($\times 10^{-3}$)	std ($\times 10^{-3}$)	gain (%)
0.5	233.301	0.151	—
1.0	233.288	0.157	0.0056
2.0	233.268	0.160	0.0145
5.0	233.260	0.150	0.0178
10.0	233.260	0.150	0.0178
30.0	**233.258**	0.153	**0.0186**

G Additional Evaluation and Results

G.1 Evaluation on the QAPLIB benchmark

Since (3) is an instance of the quadratic assignment problem, MPS can also be evaluated on the standard library for that problem [4]. Table S4 reports MPS on twenty-four QAPLIB instances of size $12 \leq n \leq 60$ against the published best known solution for each; the gap column is the relative excess over that value in percent, so that a gap of zero means the best known value was attained exactly.

MPS attains the best known solution on most instances, with a median gap of zero and a mean gap of 0.462%. The gap is zero or near zero on the dense instances and concentrates on the `chr`, `kra` and `taia` families, which are sparse or adversarially constructed. The kernels (1) of blind vision–language matching are dense and approximately metric, which is the regime of the main comparison, and the sparse and adversarially constructed families of the library lie outside it.

G.2 Anytime behaviour

Proposition 2 guarantees that the incumbent of Algorithm 1 never decreases, and Table S5 measures how quickly it improves in practice. The solver is run at $N = 100$ on CIFAR-100, matching CLIP ViT-L/14 against all-mpnet-base-v2 on the inner-product kernels of Section F, over three draws of the vision class means, and the table reports the distortion divided by N^2, averaged over the draws, as a function of the wall-clock budget. The solver is re-run from scratch at each budget, and the *gain* column records the improvement relative to the shortest budget shown.

As we can see, the sequence is monotone in the budget, as the proposition requires, and it plateaus sharply, since about four fifths of the total gain is attained within the first two seconds and nearly all of it within five, the remaining twenty-five seconds contributing about one twenty-fifth. This

Table S6: Distortion under a longer time limit. **Bold** = best, underlined = second best.

N	vision encoder	FHG (2025) [39]	time of FHG (s)	MPS (ours)
50	CLIP/ViT-L/14	955.071	2142	**954.756**
50	ConvNeXt/CN-L-1	28.145	3061	**27.574**
50	DINOv2/ViT-L/14	131.398	2811	**131.135**
100	CLIP/ViT-L/14	3847.442	5404	**3845.992**
100	ConvNeXt/CN-L-1	154.776	5403	**150.293**
100	DINOv2/ViT-L/14	495.697	5403	**493.621**

supports the matched-budget protocol, since at this size the solver reaches its plateau long before the budget it is allotted in the main comparison expires.

G.3 Behaviour under a longer time limit

The comparisons in the main text allocate one hundred and eighty seconds per solver below one hundred and fifty classes and six hundred seconds above. Since the dual solver is the slower method, we repeat the comparison under a substantially longer limit, five thousand four hundred seconds per solver, the limit adopted in the literature at larger scale, again giving our method exactly the wall-clock time the dual solver consumed.

Table S6 reports the instances obtained in this way. Our method attains the lower distortion on each, with a mean relative difference of 0.94%, larger than the difference observed under the shorter budgets, which is consistent with our method being anytime and continuing to improve when given additional time, whereas the dual ascent does not. The instances at fifty classes are informative in a different way, since there the dual solver terminated after roughly two fifths to three fifths of the time available to it, because its ascent converged before the limit, so no additional budget can alter its result at that size, and it established optimality even so on none of them. Additional time therefore leaves the picture unchanged at the sizes where it was tested.

G.4 The dual lower bound

Table S7 records how far the dual solver's bound is from being able to prove optimality. Since the Gromov–Wasserstein distortion is non-negative by construction, any lower bound that is itself non-positive excludes nothing and conveys no information about the location of the optimum. We report both the fraction of instances whose bound is non-positive and the mean relative gap between the bound and the attained solution. That gap is normalized by the attained solution,

$$\text{gap} \;=\; 100 \times \frac{\text{UB} - \text{LB}}{|\text{UB}|},$$

where UB is the distortion of the best permutation found and LB the solver's lower bound. Normalizing by UB keeps the quantity defined and finite in the regime that concerns us, since the conventional $(\text{UB} - \text{LB})/|\text{LB}|$ is undefined at $\text{LB} = 0$ and changes sign around it. A reading above one hundred percent therefore indicates a negative bound, and the excess over one hundred measures how far below zero it has fallen, in units of the attained distortion.

Figure 1 presents the same measurements in three views as a function of problem size, at left the mean relative difference in distortion between the two methods, at centre the fraction of instances on which the dual bound is non-positive, and at right the fraction on which no solution is returned within twice the allotted budget. We plot the share of non-positive bounds, since the cost-to-bound ratio used elsewhere is undefined once the bound turns non-positive.

At thirty classes the bound is informative, with a mean duality gap of three percent. The fraction of instances with a non-positive bound then grows with the problem size, and from one hundred and ten classes onward it is every instance we ran, while the mean gap grows from a few percent to several hundred. The bound thus becomes uninformative well before the solver stops returning solutions.

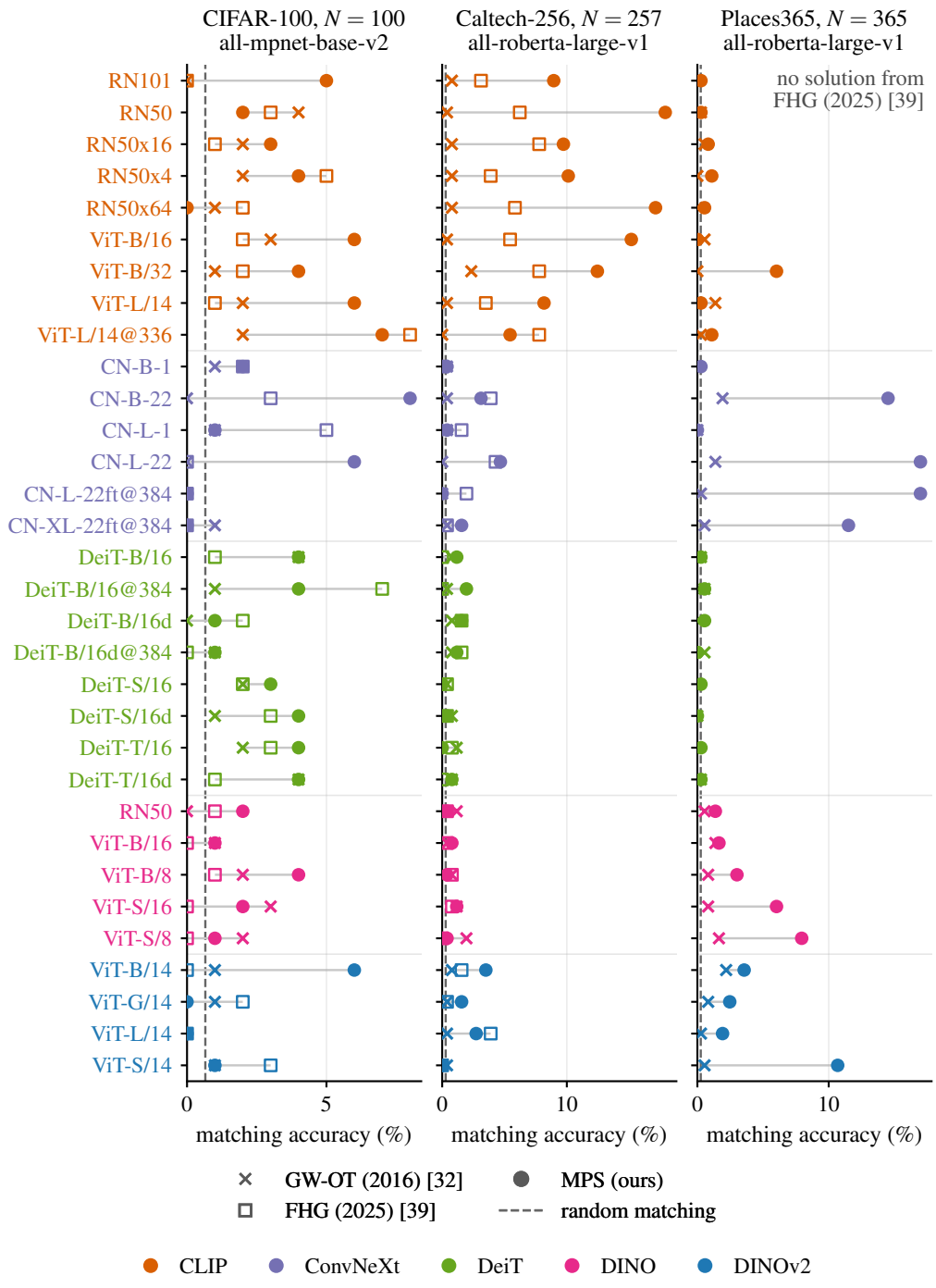


Figure S1: Matching accuracy of each vision encoder at the full class set of each dataset.

Table S7: Quality of the dual lower bound.

N	instances	non-positive bound (%)	duality gap (%)
30	60	0	3
40	36	0	4
50	60	0	5
60	36	19	103
70	36	61	405
75	36	83	928
80	36	97	576
90	36	97	568
100	60	90	700
110	24	100	380
120	24	100	374
130	24	100	392
140	24	100	395
150	24	100	394
160	6	100	435
170	3	100	336
180	6	100	406
190	3	100	313
200	24	100	742
210	3	100	300
220	6	100	428
230	3	100	351
240	5	100	416
250	24	100	767
257	3	100	351
260	1	100	288
300	9	100	1207

Table S8: Mean distortion over the thirty-two vision encoders at the full class set of each dataset (↓). Shaded row is ours. **Bold** = best, underlined = second best.

	dataset		
method	CIFAR-100	Caltech-256	Places365
classes	100	257	365
Random	1598.9	6542.9	13306.9
FAQ (2015) [46]	1525.2	6319.2	11813.9
GW-OT (2016) [32]	1502.1	6259.4	11470.8
LocalCKA (2024) [24]	1525.2	6319.2	11813.9
FHG (2025) [39]	1466.9	6080.4	—
MPS (ours)	**1462.9**	**6051.4**	**10665.9**

G.5 The model zoo at complete class sets

Table S8 and Figure S1 support the zoo-wide claims made in Section 5.4, over all thirty-two vision encoders at the complete class set of each dataset, with one language encoder per dataset. Table S8 gives the mean distortion over the encoders, and on each encoder individually our solver attains a lower distortion than every other method run on the zoo, the smallest excess of any of them over ours being 0.013%. Figure S1 gives the matching accuracy. Colour denotes the pre-training family and marker shape the solver, the thin rule spans the solvers on a given encoder, and the dashed line is the mean accuracy of a random assignment; encoders are ordered identically in every panel so that one may be followed across datasets. Each panel holds a single measurement per encoder, because at N equal to the number of classes the subset is the entire class set and there is nothing left to sample, and the panels are therefore shown without error bars. Our solver is the most accurate method, alone

or jointly, on a majority of the encoders in every dataset, and at three hundred and sixty-five classes the dual solver returns no solution on any encoder, which is why its markers are absent from the Places365 panel.

G.6 Complete results at every problem size

Table S9 extends Table 1 to every problem size we evaluated, and Table S10 gives the corresponding matching accuracies alongside. Each column is the complete case among whichever solvers have data at that size, which is what allows the table to continue past the ceiling of the earlier-stopping methods to $N = 365$.

Read across the full range, the distortion table admits no exception, since our solver attains the lowest value at every size evaluated. Second place changes hands as the range extends. The dual solver holds it at nearly every size up to two hundred and fifty-seven classes, 2-opt from two hundred and sixty to three hundred, and BAPG beyond, where neither of the two returns a solution. The ordering among the primal heuristics themselves is stable throughout, which allows the comparison to be read as a property of the solvers, independent of the class subsets drawn at each size.

In matching accuracy our solver is the most accurate method at more sizes than any other, and first or second at all but a few, with the dual solver taking most of the remaining first places. Where neither leads, the lead passes to a method that has placed a few additional classes correctly, among them FAQ and 2-opt, a scatter to be expected when the accuracy at a single size rests on few instances (Section 5.4).

G.7 Running time

Table S11 reports the mean wall-clock time consumed by each solver. The comparison is budgeted, so these figures reflect the allocation, with two exceptions. The primal heuristics terminate well before their budget, since they converge to a local optimum and stop, and the dual solver exceeds its nominal budget at larger sizes for the reason given in Section 5.1.

Read against Table S9, the table says what the budget buys. The methods that finish in a few seconds do so because they have reached a point they have no means of leaving, and each of them returns a higher distortion at every size, so the time our solver spends is what separates the two outcomes. Our own row tracks the allocation set by the dual solver, which is what makes the comparison a matched one, and it is the only method here that consumes its budget while holding a feasible matching at every moment.

G.8 Per-dataset breakdown

Table S12 gives, for each dataset, the number of classes it contains and the mean distortion attained by each solver over all sizes evaluated on it. We report the datasets separately because their distortion values differ by an order of magnitude, and an average over all three would be dominated by the datasets that happen to contribute.

The ordering is the same on all three. Our solver attains the lowest mean distortion on object categories, on general images and on scenes alike, with the dual solver second in each case and the primal heuristics clustered above both, so the advantage reported throughout this paper holds on every corpus.

Table S9: Gromov–Wasserstein distortion (↓) at every problem size evaluated. Shaded row is ours. **Bold** = best, underlined = second best.

	problem size N							
method	30	40	50	60	70	75	80	90
Random	119.1	210.7	331.9	490.6	660.3	—	870.5	1096.5
FAQ (2015) [46]	115.3	202.5	320.4	471.1	634.7	—	838.5	1056.8
GW-OT (2016) [32]	114.1	200.5	316.5	465.9	626.2	—	825.6	1042.1
2-opt (2019) [8]	112.0	197.5	312.1	455.3	613.6	—	811.2	1021.7
BAPG (2023) [21]	112.9	199.0	312.1	459.0	615.9	—	814.8	1027.3
LocalCKA (2024) [24]	115.3	202.5	320.4	471.1	634.7	—	838.5	1056.8
FHG (2025) [39]	111.1	196.3	309.1	453.3	610.7	857.0	806.1	1018.1
MPS (ours)	**111.0**	**196.2**	**308.4**	**452.4**	**609.2**	**855.8**	**803.7**	**1015.5**

	problem size N							
method	100	110	120	130	140	150	160	170
Random	1352.0	1331.1	1584.4	1882.5	2143.4	2485.7	2783.0	2849.5
FAQ (2015) [46]	1303.3	1286.4	1533.3	1814.7	2062.9	2386.7	2676.5	2804.1
GW-OT (2016) [32]	1283.5	1261.1	1507.1	1772.1	2021.7	2349.1	2636.2	2789.3
2-opt (2019) [8]	1259.6	1230.5	1469.6	1732.9	1970.7	2279.7	2564.9	2737.8
BAPG (2023) [21]	1264.3	1238.6	1474.9	1740.1	1985.1	2303.3	2589.3	2752.2
LocalCKA (2024) [24]	1303.3	1286.4	1533.3	1814.7	2062.9	2386.7	2676.5	2804.1
FHG (2025) [39]	1253.7	1225.6	1458.4	1723.2	1969.5	2275.9	2562.3	2733.7
MPS (ours)	**1249.2**	**1220.2**	**1453.0**	**1714.0**	**1954.9**	**2263.4**	**2548.4**	**2722.2**

	problem size N								
method	180	190	200	210	220	230	240	250	257
Random	3604.5	3538.7	4393.8	4332.8	5401.0	5184.3	6501.5	6134.7	6473.9
FAQ (2015) [46]	3478.4	3472.8	4234.0	4260.4	5202.2	5090.3	6242.9	6024.8	6356.6
GW-OT (2016) [32]	3397.4	3456.1	4163.8	4235.8	5079.3	5057.2	6102.9	5984.6	6314.1
2-opt (2019) [8]	3322.8	3398.0	4045.8	4172.2	4943.2	4967.2	5931.6	5891.9	6210.2
BAPG (2023) [21]	3336.6	3418.3	4077.1	4191.8	4987.3	5002.0	5986.9	5924.2	6251.9
LocalCKA (2024) [24]	3478.4	3472.8	4234.0	4260.4	5202.2	5090.3	6242.9	6024.8	6356.6
FHG (2025) [39]	3304.2	3397.5	4033.3	4162.0	4954.1	4966.6	5948.8	5887.4	6209.9
MPS (ours)	**3303.5**	**3385.2**	**4015.5**	**4148.4**	**4915.7**	**4947.5**	**5894.1**	**5863.0**	**6190.6**

	problem size N						
method	260	280	300	320	340	360	365
Random	13988.3	9771.1	11146.6	12719.1	14326.9	16070.9	16544.1
FAQ (2015) [46]	13508.0	9313.3	10539.0	11907.9	13597.9	15003.0	15691.9
GW-OT (2016) [32]	13340.6	8972.4	10220.4	11678.5	13139.1	14739.8	15151.7
2-opt (2019) [8]	13115.5	8678.1	9918.6	—	—	—	—
BAPG (2023) [21]	13266.9	8763.3	10004.2	11377.3	12849.5	14360.1	14761.7
LocalCKA (2024) [24]	13508.0	9313.3	10539.0	11907.9	13597.9	15003.0	15691.9
FHG (2025) [39]	13169.4	—	9925.4	—	—	—	—
MPS (ours)	**13090.3**	**8601.2**	**9826.4**	**11206.9**	**12596.2**	**14141.6**	**14528.9**

Table S10: Matching accuracy, in percent (↑). Shaded row is ours. **Bold** = best, underlined = second best.

	problem size N							
method	30	40	50	60	70	75	80	90
Random	6.67	2.50	5.33	1.67	1.43	—	0.83	0.00
FAQ (2015) [46]	6.67	4.17	2.67	3.33	0.48	—	0.00	0.37
GW-OT (2016) [32]	6.67	0.83	1.33	2.22	0.00	—	1.25	0.74
2-opt (2019) [8]	3.33	3.54	1.83	2.64	1.90	—	1.56	1.02
BAPG (2023) [21]	3.89	2.08	2.67	2.08	1.79	—	1.35	1.20
LocalCKA (2024) [24]	6.67	4.17	2.67	3.33	0.48	—	0.00	0.37
FHG (2025) [39]	9.33	**5.83**	5.60	**4.91**	3.77	3.56	2.53	2.90
MPS (ours)	**10.39**	5.42	**7.17**	3.80	**4.05**	**5.52**	**3.85**	**3.98**

	problem size N							
method	100	110	120	130	140	150	160	170
Random	0.00	0.61	1.11	0.51	0.71	0.67	0.42	0.98
FAQ (2015) [46]	2.00	0.30	1.11	0.77	0.95	0.44	0.83	1.37
GW-OT (2016) [32]	1.00	0.91	1.39	1.28	0.24	0.89	0.83	0.59
2-opt (2019) [8]	2.67	0.91	**3.06**	2.05	0.36	3.00	0.94	0.00
BAPG (2023) [21]	1.08	1.21	0.14	0.77	0.48	0.56	0.83	0.20
LocalCKA (2024) [24]	2.00	0.30	1.11	0.77	0.95	0.44	0.83	1.37
FHG (2025) [39]	2.00	2.20	1.87	3.53	2.32	3.50	**2.29**	3.14
MPS (ours)	**3.60**	**2.69**	2.88	**4.07**	**2.83**	**4.36**	1.15	**8.04**

	problem size N								
method	180	190	200	210	220	230	240	250	257
Random	0.37	0.53	0.50	0.63	0.23	0.72	0.28	0.13	0.26
FAQ (2015) [46]	0.37	0.53	0.75	**0.95**	0.76	0.58	0.49	0.67	0.65
GW-OT (2016) [32]	0.19	0.53	1.00	**0.95**	0.61	0.72	0.42	0.53	0.26
2-opt (2019) [8]	**1.39**	0.35	0.75	0.79	0.38	0.14	0.21	0.67	0.26
BAPG (2023) [21]	0.28	0.70	0.58	0.32	0.30	0.29	0.56	0.40	0.39
LocalCKA (2024) [24]	0.37	0.53	0.75	**0.95**	0.76	0.58	0.49	0.67	0.65
FHG (2025) [39]	0.46	**2.46**	2.75	**0.95**	**1.06**	2.17	**2.92**	**2.93**	1.95
MPS (ours)	0.56	0.53	**5.40**	0.63	0.45	**6.09**	1.18	1.93	**2.98**

	problem size N						
method	260	280	300	320	340	360	365
Random	0.51	0.36	0.33	0.10	**0.39**	0.28	0.09
FAQ (2015) [46]	0.38	**0.83**	0.33	0.21	0.29	**0.46**	0.73
GW-OT (2016) [32]	0.38	0.36	0.22	0.10	0.20	0.28	0.27
2-opt (2019) [8]	0.26	0.36	1.56	—	—	—	—
BAPG (2023) [21]	0.38	0.12	0.11	**0.42**	0.29	0.19	0.18
LocalCKA (2024) [24]	0.38	**0.83**	0.33	0.21	0.29	**0.46**	0.73
FHG (2025) [39]	0.00	—	**3.22**	—	—	—	—
MPS (ours)	**1.15**	0.36	2.08	0.31	**0.39**	0.28	**1.55**

Table S11: Wall-clock time consumed, in seconds. Shaded row is ours. **Bold** = best, underlined = second best.

	problem size N							
method	30	40	50	60	70	75	80	90
Random	4.1	4.2	4.2	4.2	4.3	—	5.3	6.4
FAQ (2015) [46]	4.1	4.3	4.1	4.2	4.8	—	5.3	6.5
GW-OT (2016) [32]	4.1	4.3	4.2	4.3	4.3	—	5.5	6.7
2-opt (2019) [8]	**2.9**	**2.9**	3.1	3.5	4.0	—	4.9	6.1
BAPG (2023) [21]	**2.9**	**2.9**	**2.9**	**2.9**	**3.0**	—	**3.0**	**3.1**
LocalCKA (2024) [24]	4.1	4.3	4.1	4.2	4.3	—	5.3	6.4
FHG (2025) [39]	182.6	184.7	182.2	183.7	184.4	**180.7**	185.0	186.0
MPS (ours)	182.6	184.7	182.3	183.7	184.4	**180.7**	185.0	186.0

	problem size N							
method	100	110	120	130	140	150	160	170
Random	6.9	7.3	7.2	8.1	7.5	8.3	4.2	9.4
FAQ (2015) [46]	6.2	7.6	7.5	7.9	8.7	8.4	5.3	10.6
GW-OT (2016) [32]	6.3	8.3	7.7	7.8	8.3	8.0	4.8	8.7
2-opt (2019) [8]	8.1	11.6	14.8	20.7	28.5	32.3	52.8	47.6
BAPG (2023) [21]	**3.1**	**3.2**	**3.3**	**3.4**	**3.4**	**3.5**	**3.6**	**3.7**
LocalCKA (2024) [24]	5.6	8.4	7.6	7.1	8.1	8.9	4.6	8.6
FHG (2025) [39]	186.8	191.2	195.6	199.9	216.2	232.2	659.0	631.3
MPS (ours)	186.8	191.2	195.6	199.9	216.2	232.3	659.0	631.3

	problem size N								
method	180	190	200	210	220	230	240	250	257
Random	5.9	9.0	7.7	8.9	4.3	5.4	**3.5**	4.5	5.7
FAQ (2015) [46]	6.8	8.6	6.6	10.9	**3.9**	4.7	3.6	4.5	5.5
GW-OT (2016) [32]	8.1	14.2	6.9	9.9	5.9	4.8	3.6	4.4	6.4
2-opt (2019) [8]	69.2	99.9	121.0	146.3	192.9	218.0	361.1	363.4	385.7
BAPG (2023) [21]	**3.8**	**3.9**	**3.9**	**4.0**	4.1	**4.2**	4.3	**4.3**	**4.3**
LocalCKA (2024) [24]	8.1	10.4	6.5	10.5	4.4	6.5	**3.5**	4.4	5.8
FHG (2025) [39]	697.3	676.6	690.7	875.8	786.3	960.1	961.9	837.2	823.6
MPS (ours)	717.2	676.7	690.7	875.9	786.5	960.3	861.3	837.2	824.2

	problem size N						
method	260	280	300	320	340	360	365
Random	**2.9**	**2.9**	**2.9**	**2.9**	**2.9**	**2.9**	**2.9**
FAQ (2015) [46]	**2.9**	**2.9**	**2.9**	**2.9**	**2.9**	**2.9**	3.0
GW-OT (2016) [32]	3.0	3.0	3.0	3.7	**2.9**	3.0	3.1
2-opt (2019) [8]	548.7	898.5	1056.8	1202.6	1202.6	1202.7	1202.7
BAPG (2023) [21]	4.3	4.6	4.7	4.8	5.0	5.9	9.0
LocalCKA (2024) [24]	**2.9**	**2.9**	**2.9**	3.3	**2.9**	**2.9**	3.0
FHG (2025) [39]	1187.9	1203.9	1145.5	1204.5	1203.9	1203.9	1202.7
MPS (ours)	785.4	600.1	994.9	600.1	600.2	601.1	600.1

Table S12: Distortion by dataset, over all sizes evaluated. Shaded row is ours. **Bold** = best, underlined = second best.

	dataset		
method	CIFAR-100	Caltech-256	Places365
classes	100	257	365
Random	759.9	2473.2	2907.6
FAQ (2015) [46]	731.5	2429.1	2761.1
GW-OT (2016) [32]	724.1	2414.8	2680.0
2-opt (2019) [8]	712.5	2375.1	2596.3
BAPG (2023) [21]	715.1	2388.7	2620.2
LocalCKA (2024) [24]	731.5	2429.1	2761.1
FHG (2025) [39]	709.8	2372.7	2592.5
MPS (ours)	**708.5**	**2364.2**	**2572.9**