ONLINE GEOMETRIC OPTIMIZATION VIA PROJECTION, HIERARCHICAL DISCRETIZATION, AND CONVEX RELAXATION

Anonymous authors
Paper under double-blind review

ABSTRACT

We study online optimization problems in which a learner must choose a compressed representation before observing each incoming data point. We focus on two settings: online k-means clustering and online weighted low-rank approximation (WLRA). For online k-means, we revisit multiplicative-weights methods over discretized experts and show how marginal Johnson–Lindenstrauss projections reduce the effective dimension without requiring uniform distance preservation over the entire stream. We further decouple projected Voronoi assignment from ambient center recovery by maintaining Follow-the-Leader centroids directly in $\mathbb{R}^d$, avoiding pseudo-inverse distortion. For online WLRA, we formulate the rank-k comparator class as the Grassmannian $\mathrm{Gr}(k, d)$ and develop a hierarchical region decomposition coupled with tree-structured multiplicative weights. This yields sublinear regret against the best fixed rank-k subspace, while making explicit the exponential cost of Grassmannian discretization. A second Grassmannian algorithm obtains a $(1 + \varepsilon)$ comparison using a number of regions independent of the horizon. We then introduce a Fantope relaxation that gives polynomial-time online updates and sublinear regret for a weighted projection surrogate against a reduced fractional comparator. Together, these results give a unified view of online approximation under nonconvex geometric structure, supported by empirical evaluations on online low-rank approximation.

1 INTRODUCTION

Modern applications generate unlabeled data at massive scales, driven in part by the shift of everyday activities to digital platforms. In *unsupervised learning*, the goal is to discover useful patterns directly from these observations: which points are similar, what features vary together, and whether a simpler representation can capture the structure of the data (Ghahramani, 2003). Such representations can make large datasets easier to explore and provide a basis for subsequent analysis. The challenge grows however when data arrives continuously. Rather than examining a complete dataset at once, a learner must use the observations available so far to build a representation that can accommodate new ones as they arrive (Aggarwal et al., 2003). This makes efficient, continually updated representations particularly valuable.

Two fundamental approaches for unsupervised learning are k-means clustering and low rank approximation. By representing observations with a small set of centers, k-means provides a simple way to group similar observations together (MacQueen, 1967; Lloyd, 1982). Low-rank approximation instead captures shared structure in a small number of factors, significantly reducing the storage and computation needed to work with high-dimensional data (Eckart & Young, 1936). These advantages have made both methods pervasive across machine learning: k-means-style clustering has supported applications ranging from image segmentation, where similar pixels are grouped into coherent regions (Achanta et al., 2012), to signal quantization, where samples are represented by a small set of prototype values (Lloyd, 1982); while low-rank structure has powered applications ranging from recommendation systems, e.g. Netflix Prize (Bell et al., 2007; 2008; Koren, 2009; Bell et al., 2010), to modern foundation models, where LoRA factors weight updates into low-rank matrices to reduce the number of trainable parameters (Hu et al., 2022; Xu et al., 2024; Wu et al., 2024; Li et al., 2024).

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.

In this work, we study k-means and a weighted variant of low-rank approximation (WLRA) in an *online* setting. At each round, the learner selects a representation before observing the next data point and incurring a loss based on that choice. Performance of the learner is then measured by the *regret*: the difference between the learner's cumulative loss and that of the best fixed representation chosen with knowledge of the full stream.

Formally, in the online k-means setting, the learner chooses a set of centers $C_t = \{c_{t,1}, \ldots, c_{t,k}\} \subseteq \mathbb{R}^d$ before observing $x_t \in \mathbb{R}^d$, and then incurs loss

$$\ell_t(C_t) := \min_{c \in C_t} \|x_t - c\|_2^2.$$

The goal is to achieve small regret relative to the best fixed set of k centers chosen after the entire stream is known. This differs from the classical streaming model, where an algorithm processes the data sequentially but is evaluated only on the final clustering it produces. In the online-learning setting, the learner incurs loss at every round.

In online WLRA, a learner must choose a rank-k subspace $U_t \in \mathrm{Gr}(k, d)$ before observing a vector $x_t \in \mathbb{R}^d$ and a weight matrix W_t. Writing $X_{U_t} \in \mathbb{R}^{d \times k}$ for some matrix having columns spanning U_t, its loss is the best weighted reconstruction error within the selected subspace:

$$\ell_t(U_t) := \min_{a \in \mathbb{R}^k} \left\| W_t^{1/2}(x_t - X_{U_t} a) \right\|_2^2.$$

The comparator is the best fixed rank-k subspace in hindsight. The weights allow different coordinates to have different importance or confidence, as in applications of weighted low-rank models to collaborative filtering and computer vision (Srebro & Jaakkola, 2003; Gillis & Glineur, 2011), but they also make the problem computationally difficult. In particular, offline WLRA is NP-hard to solve to inverse-polynomial additive accuracy, even in the rank-one case (Gillis & Glineur, 2011). This creates a tension between competing with the exact rank-k optimum and obtaining an efficient online algorithm.

A natural approach for seeking low regret in online problems is to view each candidate solution as an expert and apply multiplicative weights with techniques such as the Multiplicative Weights Update Algorithm (MWUA) (Arora et al., 2012), incurring regret $O(\sqrt{T \log n})$, where n is the number of experts (Gravin et al., 2017). The difficulty with this approach is that the space of possible choices is continuous and nonconvex in both problems, so there are infinitely many candidate experts. One can discretize these classes, but a naive discretization may contain exponentially many experts, with the exponent depending on the ambient dimension, the intrinsic dimension of the hypothesis space, or the desired approximation scale. This leads to the central question of the paper: how can geometric structure be used to obtain meaningful online regret guarantees without paying the full cost of a naive discretization?

Our results address this question differently in the two settings. For online k-means, we build on the discretization and multiplicative-weights framework of Cohen-Addad et al. (2021), in which candidate center sets are treated as experts. We combine this framework with dimension reduction while maintaining the actual centers directly in the ambient space. A marginal Johnson–Lindenstrauss argument shows that, at each round, it is sufficient to preserve the distances from the incoming point to the k optimal centers. The projected dimension therefore depends on k and the approximation parameter. Projected grid points determine Voronoi assignments, while ambient Follow-the-Leader updates maintain the corresponding centers in $\mathbb{R}^d$.

For online WLRA, we first compete with the best fixed rank-k subspace in hindsight. The natural comparator space is the Grassmannian $\mathrm{Gr}(k, d)$, whose points represent subspaces independently of their choice of basis. We cover this space with bounded coordinate charts, refine the charts into dyadic regions, and apply Mass Tree MWUA to rank-k representatives. A second-order analysis at the optimal comparator permits resolution $T^{-1/4}$ rather than $T^{-1/2}$, reducing the number of experts from $T^{k(d-k)/2}$ to $T^{k(d-k)/4}$ while retaining sublinear expected regret. We also obtain a $(1+\varepsilon)$-comparator guarantee by running a moving learner within each sufficiently small region and using MWUA to select among them. Every iterate remains a rank-k subspace, and the number of regions is independent of T. The expected cumulative loss is at most $(1+\varepsilon)$ times the optimal rank-k loss, plus $\varepsilon W\sqrt{T}$ and a sublinear online-learning term.

Both Grassmannian algorithms remain exponential in the intrinsic dimension $k(d-k)$. An online-to-offline reduction shows that a polynomial-time online algorithm with sufficiently small regret against the exact comparator would yield a randomized polynomial-time algorithm for an NP-hard offline accuracy target. The reduction also applies to a uniform $(1+\varepsilon)$ guarantee at inverse-polynomial ε, but does not establish hardness for fixed constant ε. To obtain polynomial-time updates, we instead optimize over a reduced Fantope. This method uses a weighted projection loss in place of the standard weighted reconstruction loss and competes with fractional projectors supported on an anchor subspace. Its sublinear regret guarantee is therefore for a different loss and comparator class.

Paper organization. Section 2 summarizes our theoretical and empirical contributions. Section 3 presents the online k-means algorithm, including the marginal JL argument and ambient Follow-the-Leader tracking. Section 4 presents the Grassmannian exact-comparator and $(1+\varepsilon)$-comparator algorithms, followed by the polynomial-time Fantope surrogate; the appendix gives the corresponding computational barrier for standard WLRA. Section 5 evaluates the low-rank methods on MNIST and MovieLens, with additional synthetic weighted experiments and limitations in the appendix.

2 Our Contributions

We make the following contributions for online geometric approximation, combining multiplicative weights with projection, hierarchical discretization, and convex relaxation.

Online k-means via marginal JL projections. Building on the discretized-expert framework of Cohen-Addad et al. (2021), we use a marginal Johnson–Lindenstrauss argument to construct the expert class in a reduced dimension. At fixed ε, this replaces d in the exponent of a direct kd-dimensional center grid with a projected dimension depending only on k and ε; the expert class has no explicit dependence on d or T, although its updates still process d-dimensional points. Theorem 3.1 gives the resulting expected-loss guarantee.

Online WLRA via Grassmannian algorithms. We give two algorithms for standard weighted low-rank approximation that compete with rank-k subspaces in $\mathrm{Gr}(k,d)$. Bounded graph charts let us discretize $k(d-k)$ coordinates rather than all dk entries of a basis matrix. Our first algorithm refines these charts into dyadic regions and applies Mass Tree MWUA to fixed representatives. A second-order analysis at the cumulative optimum permits resolution $T^{-1/4}$ instead of the $T^{-1/2}$ given by a first-order bound. This reduces the horizon-dependent expert count from $T^{k(d-k)/2}$ to $T^{k(d-k)/4}$ while achieving sublinear expected regret against the exact comparator (Theorem 4.1).

Our second algorithm runs a moving learner within each sufficiently small chart region and uses MWUA to select among regions. For positive diagonal weights, a locally convex quadratic surrogate yields expected loss at most $(1+\varepsilon)$ times that of the best fixed rank-k subspace, plus sublinear additive terms (Theorem 4.2). Its number of regions is independent of T at fixed ε. Both algorithms play rank-k subspaces, but their cost remains exponential in $k(d-k)$.

Online-to-offline computational barrier. Our reduction shows that a polynomial-time online algorithm with exact-comparator regret $cT^{1-\alpha}(\log(eT))^b$, where $\alpha, b > 0$ are fixed and c is horizon-independent and polynomial in the instance parameters, would yield a randomized polynomial-time algorithm for inverse-polynomial additive accuracy in offline WLRA (Theorem G.2 and corollary G.3). The positive-weight hardness result of Gillis & Glineur (2011) rules this out unless $\mathrm{NP} \subseteq \mathrm{BPP}$. The barrier also applies to uniform $(1+\varepsilon)$ guarantees at inverse-polynomial ε.

Polynomial-time surrogate for online WLRA. Motivated by this barrier, we replace the standard weighted reconstruction loss with a weighted projection loss and the exact rank-k comparator with a fractional projection model over the Fantope. Restricting the fractional projector to an anchor subspace of dimension m gives an exact reduced expression for the surrogate loss and its gradient. This leads to a polynomial-time online gradient method with per-round runtime $O(m^2d + m^3)$ and sublinear regret against the best fixed fractional comparator in the reduced Fantope (Theorem 4.3). Its guarantee concerns this surrogate loss and comparator class, rather than the standard-WLRA optimum.

Empirical evaluation. We evaluate a practical hierarchical implementation and Fantope OGD on MNIST and MovieLens, with additional weighted synthetic experiments. The hierarchy substantially improves over a fixed-subspace baseline and performs competitively with the adaptive base-

lines considered. Fantope OGD also achieves competitive cumulative projection loss on MNIST; on synthetic streams with unequal coordinate reliability, weighting improves its recovery of the generating subspace relative to its unweighted version and streaming SVD. The real-data experiments are unweighted, and the practical hierarchy differs from the fixed-depth construction analyzed in our theorem.

3 Online k-Means

We first present a finite-dimensional version of our approach in the online k-means setting. Following the online k-means formulation of Cohen-Addad et al. (2021), at each round t the learner chooses a set of centers $C_t = \{c_{t,1}, \dots, c_{t,k}\} \subset \mathbb{R}^d$ before observing x_t, and then incurs $\ell_t(C_t) := \min_{c \in C_t} \|x_t - c\|_2^2$.

Let $C^\star = \{c_1^\star, \dots, c_k^\star\}$ denote the optimal fixed set of centers in hindsight, and write $\mathcal{C}_{\mathrm{OPT}} := \sum_{t=1}^T \min_{j \in [k]} \|x_t - c_j^\star\|_2^2$. We assume throughout this section that $\|x_t\|_2 \le 1$ and $\|c_j^\star\|_2 \le 1$. We also assume the stream is chosen independently of the random projection used below. In particular, both $x_{1:T}$ and its hindsight comparator $C^\star$ are fixed before the projection is sampled. The same analysis applies to a random stream whose distribution is independent of the projection, by conditioning on the realized stream.

Discretizing k centers directly in $\mathbb{R}^d$ gives an expert class exponential in kd. A Johnson–Lindenstrauss (JL) projection offers a smaller space in which to specify the experts, but requiring one projection to preserve all relevant distances simultaneously over T rounds introduces a $\log T$ term in its dimension. Here we need a weaker guarantee. The projected optimal centers need to lie in the grid box, an event whose probability can be controlled using the k fixed centers. For each fixed round, we additionally compare the k distances from x_t to those centers. Neither bound requires a union bound over the T rounds.

3.1 Projected Voronoi Experts with Ambient FTL Tracking

Fix $0 < \varepsilon \le 1/2$. Let $\Pi \in \mathbb{R}^{m \times d}$ have independent entries distributed as $\mathcal{N}(0, 1/m)$, where $m = \lceil C\varepsilon^{-2} \log \frac{k}{\varepsilon} \rceil$ for a sufficiently large universal constant C. Let S_m be an ε-net of the box $[-2, 2]^m$ with mesh width $\varepsilon/\sqrt{m}$, so that every projected point in the box is within Euclidean distance at most ε of some grid point. We define each expert as a tuple $Y = (y_1, \dots, y_k) \in S_m^k$. The grid points y_j are not centers in $\mathbb{R}^d$, rather, they only define Voronoi cells in the projected space. Each expert maintains ambient statistics $\left(N_{t,j}^Y, \mu_{t,j}^Y\right)_{j=1}^k$, where $N_{t,j}^Y$ is the number of points previously routed to cluster j by expert Y, and $\mu_{t,j}^Y \in \mathbb{R}^d$ is their empirical mean.

At round t, MWUA samples an expert Y_t from its current distribution and outputs the current means, $C_t(Y_t) := \{\mu_{t,1}^{Y_t}, \dots, \mu_{t,k}^{Y_t}\}$. After observing x_t, every expert Y assigns x_t according to projected Voronoi distance, $a_t^Y := \arg\min_{j \in [k]} \|\Pi x_t - y_j\|_2^2$. The expert is then charged for predicting with the *ambient* mean of the assigned cluster, $\widetilde{\ell}_t(Y) := \|x_t - \mu_{t,a_t^Y}^Y\|_2^2$. This charge upper-bounds the learner's actual loss when Y is sampled, because $\ell_t(C_t(Y)) = \min_{j \in [k]} \|x_t - \mu_{t,j}^Y\|_2^2 \le \widetilde{\ell}_t(Y)$. Finally, the expert updates the count and mean of cluster a_t^Y:

$$N_{t+1,a_t^Y}^Y = N_{t,a_t^Y}^Y + 1, \qquad \mu_{t+1,a_t^Y}^Y = \mu_{t,a_t^Y}^Y + \frac{x_t - \mu_{t,a_t^Y}^Y}{N_{t+1,a_t^Y}^Y},$$

and all other cluster statistics remain unchanged. Thus the projected grid controls only the assignments, while the actual centers are recovered natively in $\mathbb{R}^d$ through Follow-the-Leader means.

3.2 Analysis

The guarantee follows from three simple components, whose full statements and proofs appear in Appendix C. First, discretizing the projected optimal centers gives an expert whose routing decisions have small expected excess assignment cost. For each fixed t, the marginal JL bound controls

Algorithm 1 Projected Voronoi FTL for online k-means

Sample $\Pi \in \mathbb{R}^{m\times d}$ and construct grid $S_m \subset [-2,2]^m$.
Let $\mathcal{E} = S_m^k$. Initialize $w_1(Y) = 1$ for all $Y \in \mathcal{E}$.
Initialize $N_{1,j}^Y = 0$ and $\mu_{1,j}^Y = 0$ for all $Y \in \mathcal{E}$ and $j \in [k]$.
for $t = 1, \dots, T$ **do**
    Sample Y_t with probability proportional to $w_t(Y)$.
    Output $C_t = \{\mu_{t,1}^{Y_t}, \dots, \mu_{t,k}^{Y_t}\}$.
    Observe x_t and compute $z_t = \Pi x_t$.
    for each $Y = (y_1, \dots, y_k) \in \mathcal{E}$ **do**
        $a_t^Y \leftarrow \arg\min_{j\in[k]} \|z_t - y_j\|_2^2$.
        $\widetilde{\ell}_t(Y) \leftarrow \|x_t - \mu_{t,a_t^Y}^Y\|_2^2$.
        $w_{t+1}(Y) \leftarrow w_t(Y)\exp(-\eta\widetilde{\ell}_t(Y))$.
        Update $N_{t,a_t^Y}^Y$ and $\mu_{t,a_t^Y}^Y$ using the ambient mean update.

the k distances $\|x_t - c_j^\star\|_2$ without a union bound over the rounds. Second, for any expert's routing sequence, Follow-the-Leader means compete with fixed ambient centers for the induced clusters up to $O(k\log T)$.Finally, MWUA ensures the learner competes with the resulting expert up to $O(\sqrt{T\log|\mathcal{E}|})$.

Theorem 3.1. *Run Algorithm 1 with $\eta = \Theta(\sqrt{\log|\mathcal{E}|/T})$. Then*

$$\mathbb{E}_{\Pi,\mathrm{alg}}\left[\sum_{t=1}^{T}\ell_t(C_t)\right] \le (1+O(\varepsilon))\mathcal{C}_{\mathrm{OPT}} + O(\varepsilon T) + O\left(\sqrt{T\log|\mathcal{E}|}\right) + O(k\log T).$$

Moreover, $\log|\mathcal{E}| = O\left(km\log\frac{\sqrt{m}}{\varepsilon}\right)$ and $m = O\left(\varepsilon^{-2}\log\frac{k}{\varepsilon}\right)$, so the expert complexity is independent of both the ambient dimension d and the time horizon T.

The horizon independence should be interpreted at fixed approximation scale ε. For any fixed ε, the projected expert class has size depending only on (k,ε), not on T or d. If one subsequently tunes ε as a function of T to optimize the additive discretization term, this dependence re-enters only through the chosen accuracy parameter, not through a JL union bound over the stream.

Remark 1. *For $T \ge 16$, setting $\varepsilon = T^{-1/4}$ balances the $O(\varepsilon T)$ and $O(\sqrt{T\log|\mathcal{E}|})$ terms, since $\log|\mathcal{E}| = O(k\varepsilon^{-2}\log(k/\varepsilon))$ grows as ε^{-2}. This gives an overall regret bound of $O(T^{3/4}\cdot \mathrm{polylog}(T))$. For smaller T, one can take $\varepsilon = 1/2$.*

Dependence on the stream. Independence from Π is essential to the marginal JL argument: once x_t or $C^\star$ can depend on the sampled projection, their difference vectors are no longer fixed when the JL bound is applied. There is also a direct obstruction to extending this analysis to projection-dependent data. When $m < d$, choose a unit vector $v \in \ker(\Pi)$ and let the stream alternate between v and $-v$. Every projected point is then zero. For each expert, the routing decision is the same on every round, so its two distinct ambient clusters cannot track the two values, even though a fixed pair of centers $\{v,-v\}$ has zero loss. An adversary that learns Π, or chooses points using information about it revealed by the learner, can therefore defeat this projected-routing argument. The theorem does not claim a guarantee against such adaptive adversaries; handling them would require a different projection guarantee or algorithm.

Computational cost. The flat implementation updates every expert on every round and has per-round time $O\big(md + |\mathcal{E}|(km+d)\big) = O\big(d\cdot C(k,\varepsilon)\big)$, where $C(k,\varepsilon)$ may be exponential in k and ε^{-1} but is independent of d and T except for the linear cost of applying the projection. Thus the role of this section is conceptual rather than practical as marginal JL removes the horizon-dependent discretization barrier by separating the low-dimensional routing rule from the ambient representation of the centers.

4 Online Weighted Low-Rank Approximation

For standard WLRA, a rank-k hypothesis is a subspace rather than a particular basis, making the Grassmannian $\mathrm{Gr}(k, d)$ the natural hypothesis space. We first build a hierarchy over this space to compete with the best fixed rank-k subspace in hindsight. We then allow a $(1 + \varepsilon)$ comparison by placing a convex learner inside each sufficiently small region, obtaining a number of regions independent of the horizon. Both algorithms retain a cost exponential in the intrinsic dimension $k(d - k)$, which motivates the Fantope surrogate we introduce next for efficient online updates against a relaxed fractional comparator.

4.1 Grassmannian HRD for Online WLRA

Because weighted reconstruction loss depends on the subspace rather than its choice of basis, we identify each subspace with its orthogonal projector. The resulting hypothesis class is

$$\mathcal{P}_k(d) = \{P \in \mathbb{R}^{d\times d} : P = P^\top,\ P^2 = P,\ \mathrm{tr}\, P = k\}.$$

For $P \in \mathcal{P}_k(d)$, let X_P be any full-column-rank matrix whose columns span its range. On round t, the learner chooses P_t before seeing (x_t, W_t) and incurs the standard weighted reconstruction loss $\ell_t(P_t) = \min_{a\in\mathbb{R}^k} \|W_t^{1/2}(x_t - X_{P_t}a)\|_2^2$. We assume throughout that $1 \le k < d$, $\|x_t\|_2 \le 1$, and $\mu I_d \preceq W_t \preceq W I_d$ for $0 < \mu \le W$. Set $\kappa = W/\mu$ and assume the input sequence is oblivious. Our benchmark is

$$P^\star \in \arg\min_{P\in\mathcal{P}_k(d)} \sum_{t=1}^{T} \ell_t(P), \qquad C_{\mathrm{WLRA}} = \sum_{t=1}^{T} \ell_t(P^\star).$$

The positive lower bound on the weights is used to control how this loss changes between nearby subspaces. The exact-comparator algorithm below allows any W_t satisfying the spectral bounds; the locally convexified algorithm that follows assumes diagonal W_t.

Exact rank-k comparator. Our first algorithm constructs a finite collection of representative subspaces before the stream begins. It treats these representatives as experts: on each round, it samples one to play and then updates their weights using the observed WLRA losses. Its regret has two sources: approximating $P^\star$ by one representative and learning which representative to play.

We construct the representatives using $\binom{d}{k}$ graph charts of $\mathrm{Gr}(k, d)$. A chart selects k reference coordinates and normalizes those coordinates to the identity. The remaining coordinates form a matrix $Y \in [-1, 1]^{(d-k)\times k}$ describing the subspace. Such a chart exists for every subspace by choosing a largest-determinant $k \times k$ row minor of a basis. The appendix gives the precise construction and resolves overlaps between these charts.

In each chart, the coordinate box is divided into dyadic cubes. The center of a cube specifies a rank-k representative subspace, and halving each coordinate gives $2^{k(d-k)}$ children. We refine all charts to a common depth L before the first round and use the resulting leaf centers as experts. MTMW assigns prior mass to the leaves, samples a leaf on each round, and updates their masses after observing (x_t, W_t). The learner may play different leaves over time, but the leaf used to compare with $P^\star$ has one fixed representative.

We choose L to control the approximation cost. For subspaces U, V with principal angles $\theta_1, \ldots, \theta_k$, the chordal distance is $d_G(U, V) = \left(\sum_{i=1}^k \sin^2\theta_i\right)^{1/2} = \frac{1}{\sqrt{2}}\|P_U - P_V\|_F$. A depth-$j$ cube has chordal radius at most $\sqrt{k(d-k)}\,2^{-j}$. We set L so that the leaf representative $P_{p^\star}$ associated with $P^\star$ satisfies $\|P_{p^\star} - P^\star\|_F \le \kappa^{-1}T^{-1/4}$. This scale follows from a property of the *cumulative* loss $F(P) = \sum_t \ell_t(P)$: its first derivative vanishes at the minimizer $P^\star$. The appendix proves the resulting second-order estimate $F(P) - F(P^\star) \le \frac{3\pi^2}{2} W\kappa^2 T\|P - P^\star\|_F^2$. Hence replacing $P^\star$ by the fixed leaf representative costs at most $\frac{3\pi^2}{2} W\sqrt{T}$. The cancellation occurs only for the entire sum of losses. It would not justify using a different representative on each round, which explains the up-front refinement.

The remaining cost comes from selecting among the leaves. MTMW gives the virtual root mass 1 and splits mass equally first among charts and then among children. There are $N_{\mathrm{ex}} = \binom{d}{k}2^{k(d-k)L}$

leaves, each with mass $1/N_{\mathrm{ex}}$. Its online-learning guarantee therefore adds $W\sqrt{T\log N_{\mathrm{ex}}/2}$ to the loss of the fixed comparator leaf. Combining this term with the preceding approximation gives the exact-comparator result.

Theorem 4.1. *Under the preceding assumptions, Algorithm 4 satisfies*

$$\mathbb{E}\sum_{t=1}^{T}\ell_t(P_t) - C_{\mathrm{WLRA}} \le O\Big(W\sqrt{k(d-k)T\log(\kappa k(d-k)T)}\Big).$$

In the exact-real arithmetic model, with scalar exponentiation and finite-distribution sampling counted as primitive operations, its total running time is $O\Big(T(4\kappa\sqrt{2k(d-k)}\,T^{1/4})^{k(d-k)}\,\mathrm{poly}(d)\Big)$.

This guarantee uses the standard WLRA loss above and compares with an actual rank-k subspace. The cost of doing so is computational: the number of fixed leaf experts grows as $T^{k(d-k)/4}$ for fixed d, k, κ. The appendix supplies the depth choice, the chart and loss estimates, and the MTMW analysis.

$(1+\varepsilon)$ **rank-k comparator.** The preceding construction needs finer cells as the horizon grows because each leaf plays only its center. We can remove this dependence on T from the number of cells by letting an online learner move inside each cell. This also gives a relative comparison; if the optimum has zero loss, a nearby fixed center could have positive loss, so an additive net approximation alone cannot establish a $(1+\varepsilon)$ guarantee.

We use the same bounded graph charts. In chart J, order the coordinates as (J, J^c), put $X_J(Y) = \binom{I_k}{Y}$, and let $P_J(Y)$ project onto its columns. For diagonal W_t, we write $r_{t,J}(Y) = x_{t,J^c} - Yx_{t,J}$, and $S_{t,J}(Y) = W_{t,J^c}^{-1} + YW_{t,J}^{-1}Y^\top$. Eliminating the weighted least squares coefficients yields $\ell_t(P_J(Y)) = r_{t,J}(Y)^\top S_{t,J}(Y)^{-1} r_{t,J}(Y)$. This expression is exact, but the matrix $S_{t,J}(Y)$ varies with Y. Within a sufficiently small cube c centered at $\bar{Y}_c$, we hold that matrix fixed and optimize the surrogate $q_{t,J,c}(Y) = r_{t,J}(Y)^\top S_{t,J}(\bar{Y}_c)^{-1} r_{t,J}(Y)$. The residual is affine in Y, making $q_{t,J,c}$ a convex quadratic. Moreover, the true loss and surrogate satisfy, for every $Y \in c$ and every round,

$$\frac{q_{t,J,c}(Y)}{1+\varepsilon/4} \le \ell_t(P_J(Y)) \le \frac{q_{t,J,c}(Y)}{1-\varepsilon/4}. \tag{1}$$

A cube side length $O(\varepsilon/(\kappa k(d-k)))$ suffices, independently of T. Because the comparison is relative, it also holds at zero loss.

Each cube runs projected gradient descent on its own surrogate, with every iterate representing a genuine rank-k projector. After each round, MWUA updates the cubes using their *true* WLRA losses; the full-information setting lets us update all of their inner learners as well. For the cube containing $P^\star$, gradient descent and both sides of (1) give a $(1+\varepsilon)$ comparison plus $\varepsilon W\sqrt{T}$ additive loss. MWUA adds the cost of selecting that cube. Thus the number of cells is independent of the horizon, even though the algorithm still visits every cell on each round.

Theorem 4.2 (Horizon-independent discretization)**.** *For $0 < \varepsilon \le 1$ and diagonal W_t under the preceding spectral bounds, Algorithm 5 satisfies*

$$\mathbb{E}\sum_{t=1}^{T}\ell_t(P_t) \le (1+\varepsilon)C_{\mathrm{WLRA}} + \varepsilon W\sqrt{T} + W\sqrt{\frac{T}{2}k(d-k)\log\frac{96\kappa k(d-k)}{\varepsilon}}.$$

Its total running time in the same exact-real model is $O\big(T(48\kappa k(d-k)/\varepsilon)^{k(d-k)}(dk^2+k^3)\big)$.

The appendix specifies the cell depth and learning rates and proves the relative comparison and both online-learning bounds. In its notation, the number of cells is $N_\varepsilon = \binom{d}{k}2^{k(d-k)D}$, independent of T; memory is $O(N_\varepsilon k(d-k))$. Both constructions remain exponential in the intrinsic dimension $k(d-k)$, but they use that cost differently: the exact-comparator algorithm keeps one fixed expert per leaf, whereas the $(1+\varepsilon)$ algorithm runs one moving learner per cell.

4.2 FANTOPE RELAXATION AND POLYNOMIAL-TIME ONLINE WLRA

Given the disappointing runtime of the Grassmannian-HRD algorithms, one may wonder whether there is a way to avoid explicitly storing a weight for each of the exponentially many experts and speed up the MWUA algorithm. The result in Appendix G gives evidence that it is unlikely that a significant speed-up is possible with the stated sublinear regret guarantees, unless NP $\subseteq$ BPP. We therefore study a surrogate for online WLRA. First, we replace the standard weighted reconstruction loss, which optimizes the reconstruction coefficients within each subspace, by the weighted error of an orthogonal projection. Second, we allow fractional projectors, but restrict them to a fixed m-dimensional anchor subspace. These changes give a convex objective with polynomial-time online updates.

A rank-k orthogonal projector P_U satisfies $P_U^\top = P_U, P_U^2 = P_U, \mathrm{Tr}(P_U) = k$. Its convex hull is the Fantope

$$\mathcal{F}_k(d) = \{P \in \mathbb{R}^{d\times d} : P = P^\top,\ 0 \preceq P \preceq I_d,\ \mathrm{Tr}(P) = k\}.$$

The eigenvalues of $P \in \mathcal{F}_k(d)$ lie in $[0,1]$ and sum to k. Thus P may be viewed as a fractional rank-k projection, assigning partial mass to multiple directions.

Reduced anchor subspace. Optimizing directly in $\mathcal{F}_k(d)$ would still require manipulating $d\times d$ matrices. We therefore restrict the fractional projector to an anchor subspace. Let $V \in \mathbb{R}^{d\times m}, V^\top V = I_m$, where $k \le m \ll d$. The reduced Fantope is therefore $\mathcal{F}_k(m)$. Each reduced fractional projector $P \in \mathcal{F}_k(m)$ is lifted to the ambient space by $\nu(P) = VPV^\top$. This lifted matrix is a valid fractional projector in the ambient dimension: it is symmetric, positive semidefinite, bounded above by I_d, and has trace k: $\mathrm{Tr}(VPV^\top) = \mathrm{Tr}(PV^\top V) = \mathrm{Tr}(P) = k$. Thus the algorithm remains an online low-rank method in the ambient space, but all optimization occurs in dimension m.

The surrogate loss of a lifted fractional projector is $\widetilde{\ell}_t(P) := \|W_t^{1/2}(I_d - VPV^\top)x_t\|_2^2$. Define the reduced sufficient statistics

$$\alpha_t = x_t^\top W_t x_t, \quad \bar{x}_t = V^\top x_t, \quad \bar{z}_t = V^\top W_t x_t, \quad \widetilde{W}_t = V^\top W_t V, \quad \bar{X}_t = \bar{x}_t\bar{x}_t^\top.$$

Expanding this surrogate gives the exact reduced expression $\widetilde{\ell}_t(P) = \alpha_t - \mathrm{Tr}\left(P(\bar{z}_t\bar{x}_t^\top + \bar{x}_t\bar{z}_t^\top)\right) + \mathrm{Tr}\left(P\widetilde{W}_tP\bar{X}_t\right)$. This is an exact expression for the ambient surrogate loss using m-dimensional quantities. The anchor V restricts the comparator class; the reduction introduces no further approximation when evaluating $\widetilde{\ell}_t(P)$. For nonuniform weights, even an integral orthogonal projector generally has a different loss from the standard WLRA loss, because standard WLRA chooses the best weighted reconstruction within the subspace.

The first term α_t is constant in P, and the second term is affine. The only nontrivial term is $H_t(P) = \mathrm{Tr}(P\widetilde{W}_tP\bar{X}_t)$. Since $\bar{X}_t = \bar{x}_t\bar{x}_t^\top$, we can rewrite this as $H_t(P) = \bar{x}_t^\top P\widetilde{W}_tP\bar{x}_t = \|\widetilde{W}_t^{1/2}P\bar{x}_t\|_2^2$. Because $\widetilde{W}_t \succeq 0$, this is the squared norm of a linear map in P, and is therefore convex. Hence $\widetilde{\ell}_t(P)$ is convex over $\mathcal{F}_k(m)$. The gradient is $\nabla\widetilde{\ell}_t(P) = -(\bar{z}_t\bar{x}_t^\top + \bar{x}_t\bar{z}_t^\top) + \widetilde{W}_tP\bar{X}_t + \bar{X}_tP\widetilde{W}_t$. Thus each online update can be performed using standard Online Gradient Descent followed by projection back to the Fantope.

Algorithm 2 Polynomial-Time OGD over the Reduced Fantope

Require: Sequence $\{x_t, W_t\}_{t=1}^T$, anchor $V \in \mathbb{R}^{d\times m}$, step size η.

Initialize $\tilde{P}_1 = \frac{k}{m}I_m \in \mathcal{F}_k(m)$.
for $t = 1$ **to** T **do**
  $\tilde{x}_t \leftarrow V^\top x_t$
  $\tilde{z}_t \leftarrow V^\top W_t x_t$
  $\tilde{W}_t \leftarrow V^\top W_t V$
  $\nabla\tilde{\ell}_t(\tilde{P}_t) \leftarrow -(\tilde{x}_t\tilde{z}_t^\top + \tilde{z}_t\tilde{x}_t^\top) + \tilde{W}_t\tilde{P}_t\tilde{x}_t\tilde{x}_t^\top + \tilde{x}_t\tilde{x}_t^\top\tilde{P}_t\tilde{W}_t$
  $M_t \leftarrow \tilde{P}_t - \eta\nabla\tilde{\ell}_t(\tilde{P}_t)$
  Compute eigendecomposition $M_t = Q\Lambda Q^\top$
  Project eigenvalues Λ onto $\{\lambda \in [0,1]^m \mid \sum_i \lambda_i = k\}$ to get $\hat{\Lambda}$
  $\tilde{P}_{t+1} \leftarrow Q\hat{\Lambda}Q^\top$

Projection onto the Fantope. The projection step has a simple spectral form. Since $\mathcal{F}_k(m)$ is spectral, projecting M_t onto $\mathcal{F}_k(m)$ reduces to projecting the eigenvalue vector λ onto the capped simplex $\Delta_k^{[0,1]} = \{\mu \in [0,1]^m : \sum_{i=1}^m \mu_i = k\}$. If $\widehat{\lambda}$ is this Euclidean projection, then $\Pi_{\mathcal{F}_k(m)}(M_t) = Q\,\mathrm{diag}(\widehat{\lambda})\,Q^\top$. The dominant optimization cost is the eigendecomposition in m dimensions.

The Fantope has a bounded diameter: for any $P, Q \in \mathcal{F}_k(m)$, $\|P - Q\|_F^2 \le 2k$. Under the assumptions $\|x_t\|_2 \le 1$ and $\|W_t\|_{\mathrm{op}} \le W$, the gradients satisfy $\|\nabla \widetilde{\ell}_t(P)\|_F \le O(W)$. Applying the standard OGD regret theorem gives the following guarantee.

Theorem 4.3 (Polynomial-time fractional WLRA regret)**.** *Let V be fixed before round 1, let $\|x_t\|_2 \le 1$, and let W_t be diagonal with $0 \preceq W_t \preceq W I_d$, where $W > 0$. Running Algorithm 2 with $\eta = \Theta\left(\frac{\sqrt{k}}{W\sqrt{T}}\right)$ guarantees that for every fixed fractional comparator $P^\star \in \mathcal{F}_k(m)$, $\sum_{t=1}^T \widetilde{\ell}_t(P_t) - \sum_{t=1}^T \widetilde{\ell}_t(P^\star) \le O(W\sqrt{kT})$. Furthermore, the per-round runtime is $O(m^2 d + m^3)$.*

This bound compares surrogate losses within the reduced Fantope; it makes no multiplicative or additive comparison to C_{WLRA}. The term $O(m^2 d)$ comes from forming $\widetilde{W}_t = V^\top W_t V$; since W_t is diagonal, this can be done without forming any dense $d \times d$ matrix. The term $O(m^3)$ comes from the eigendecomposition needed for Fantope projection.

5 Experiments on MNIST

We evaluate a practical unweighted hierarchical subspace method on MNIST (LeCun et al., 1998), with implementation details and a MovieLens experiment in Appendix H. The implementation uses a simplified loss-dependent binary tree of subspace centers; it is neither the fixed-depth Grassmannian algorithm nor the locally convexified $(1+\varepsilon)$ algorithm analyzed above. We also report a synthetic study in Appendix H.3 that examines whether coordinate-dependent weights improve subspace recovery for the Fantope method. These studies provide empirical examples of the practical hierarchy and weighted Fantope update; they do not test the stated standard-WLRA regret guarantees.

Experimental setup. We use MNIST's 70,000 images with 784 features. Truncated SVD is fitted to all 70,000 images before the first streaming round; we then normalize each reduced row to unit length and evaluate on the first $T = 5000$ rows, in permuted stream orders, for $d = 50$ and $k \in \{10, 15, 20\}$. Thus, the online methods operate in a precomputed, data-dependent feature space that includes the evaluated images. The experiment does not simulate learning the feature map online from previously seen images. The supplied runner specifies ten seeded stream permutations for each rank. The hierarchy uses $n_{\min} = 10$, $n_{\max} = 100$, $\eta = 0.5$, and $\varepsilon_{\mathrm{hrd}} = 0.1$. We compare its reported cumulative residual with a fixed random control, Fantope OGD, FTRL, and an oracle rank-k projector fitted by SVD to the entire evaluated stream. The oracle uses future observations and serves as a retrospective rank-k reference, rather than an online method. The source code is available at `https://anonymous.4open.science/r/Online-Low-Rank-Approximation-042C/`.

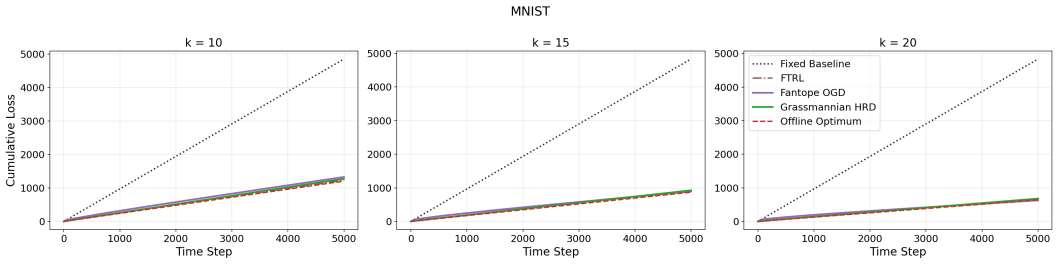

Fig. 1: Cumulative projection loss on MNIST over $T = 5000$ streamed points for $k \in \{10, 15, 20\}$.

Results. As shown in Figure 1, the hierarchical method substantially improves over the fixed baseline, reducing cumulative loss by 73.78% at $k = 10$, 80.87% at $k = 15$, and 85.93% at $k = 20$. It also performs competitively with the Fantope OGD and adaptive baseline FTRL, and remains closer to the offline optimum than the fixed baseline. These results suggest that the hierarchical method adapts to low-dimensional structure in the stream, and that the benefit over a fixed baseline becomes more pronounced as the target rank increases.

AI USE STATEMENT

The results in Section 3 and Section 4.2 were developed with assistance from Gemini DeepThink. ChatGPT and Claude assisted in refining and checking the proofs in Section 4 and Appendix G, and in locating relevant references. ChatGPT was also used to outline sections and check grammar, while Claude assisted in developing and debugging the experimental code. The authors wrote the final arguments and independently reviewed and verified the proofs, algorithms, references, code, experiments, and claims.

REFERENCES

P.-A. Absil, Robert Mahony, and Rodolphe Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, 2008. 15

Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE transactions on pattern analysis and machine intelligence*, 34(11):2274–2282, 2012. 1

Charu C Aggarwal, S Yu Philip, Jiawei Han, and Jianyong Wang. A framework for clustering evolving data streams. In *Proceedings 2003 VLDB conference*, pp. 81–92. Elsevier, 2003. 1

Sanjeev Arora, Elad Hazan, and Satyen Kale. The multiplicative weights update method: a meta-algorithm and applications. *Theory of computing*, 8(1):121–164, 2012. 2, 15

David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 1027–1035, 2007. 13

Laura Balzano, Robert Nowak, and Benjamin Recht. Online identification and tracking of subspaces from highly incomplete information. *Allerton Conference on Communication, Control, and Computing*, pp. 704–711, 2010. 13

Robert M Bell, Yehuda Koren, and Chris Volinsky. The bellkor solution to the netflix prize. *KorBell Team's Report to Netflix*, 2007. 1

Robert M Bell, Yehuda Koren, and Chris Volinsky. The bellkor 2008 solution to the netflix prize. *Statistics Research Department at AT&T Research*, 1(1), 2008. 1

Robert M Bell, Yehuda Koren, and Chris Volinsky. All together now: A perspective on the netflix prize. *Chance*, 23(1):24–29, 2010. 1

Åke Björck and Gene H. Golub. Numerical methods for computing angles between linear subspaces. *Mathematics of Computation*, 27(123):579–594, 1973. 15

Matthew Brand. Fast low-rank modifications of the thin singular value decomposition. *Linear algebra and its applications*, 415(1):20–30, 2006. 13

Nicolo Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games*. Cambridge university press, 2006. 15

Kenneth L. Clarkson. Las vegas algorithms for linear and integer programming when the dimension is small. *J. ACM*, 42(2):488–499, 1995. 15

Vincent Cohen-Addad, Benjamin Guedj, Varun Kanade, and Guy Rom. Online k-means clustering. In *The 24th International Conference on Artificial Intelligence and Statistics, AISTATS*, pp. 1126–1134, 2021. 2, 3, 4, 13, 16

Thomas M. Cover. Behaviour of sequential predictors of binary sequences. In *Proceedings of the Symposium on Information Theory, Statistical Decision Functions, Random Processes*, Prague, 1967. Czechoslovak Academy of Sciences, Publishing House of the Czechoslovak Academy of Sciences. Held in Prague, August 31 to September 11, 1965. 15

Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936. 1, 13

Alan Edelman, Tomás A. Arias, and Steven T. Smith. The geometry of algorithms with orthogonality constraints. *SIAM Journal on Matrix Analysis and Applications*, 20(2):303–353, 1998. 15

Jiashi Feng, Huan Xu, and Shuicheng Yan. Online robust pca via stochastic optimization. *Advances in Neural Information Processing Systems*, 26:404–412, 2013. 13

Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997. 15

Yoav Freund and Robert E Schapire. Adaptive game playing using multiplicative weights. *Games and Economic Behavior*, 29(1-2):79–103, 1999. 15

Dan Garber, Elad Hazan, and Tengyu Ma. Online learning of eigenvectors. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 560–568, 2015. 14

Naveen Garg and Jochen Könemann. Faster and simpler algorithms for multicommodity flow and other fractional packing problems. *SIAM J. Comput.*, 37(2):630–652, 2007. 15

Zoubin Ghahramani. Unsupervised learning. In *Summer school on machine learning*, pp. 72–112. Springer, 2003. 1

Nicolas Gillis and François Glineur. Low-rank matrix approximation with weights or missing data is np-hard. *SIAM J. Matrix Anal. Appl.*, 32(4):1149–1165, 2011. 2, 3, 14, 34

Nick Gravin, Yuval Peres, and Balasubramanian Sivan. Tight lower bounds for multiplicative weights algorithmic families. In *44th International Colloquium on Automata, Languages, and Programming, ICALP*, pp. 48:1–48:14, 2017. 2

Nathan Halko, Per-Gunnar Martinsson, and Joel A Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53 (2):217–288, 2011. 13

Sariel Har-Peled and Soham Mazumdar. On coresets for k-means and k-median clustering. In *Proceedings of the Thirty-Sixth Annual ACM Symposium on Theory of Computing*, pp. 291–300, 2004. 13

F. Maxwell Harper and Joseph A. Konstan. The movielens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems*, 5(4):19:1–19:19, 2015. doi: 10.1145/2827872. 36

Elad Hazan, Tomer Koren, Roi Livni, and Yishay Mansour. Online learning with low rank experts. In *Proceedings of the 29th Conference on Learning Theory, COLT*, pp. 1096–1114, 2016. 15

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR*, 2022. 1

Ian T Jolliffe. *Principal component analysis*. Springer, 2002. 13

Adam Tauman Kalai and Santosh S. Vempala. Efficient algorithms for online decision problems. *J. Comput. Syst. Sci.*, 71(3):291–307, 2005. 15

Tapas Kanungo, David M. Mount, Nathan S. Netanyahu, Christine D. Piatko, Ruth Silverman, and Angela Y. Wu. A local search approximation algorithm for k-means clustering. *Computational Geometry*, 28(2–3):89–112, 2004. 13

Yehuda Koren. The bellkor solution to the netflix grand prize. *Netflix prize documentation*, 81 (2009):1–10, 2009. 1

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. The mnist database of handwritten digits, 1998. URL `http://yann.lecun.com/exdb/mnist/`. 9

Yixiao Li, Yifan Yu, Chen Liang, Nikos Karampatziakis, Pengcheng He, Weizhu Chen, and Tuo Zhao. Loftq: Lora-fine-tuning-aware quantization for large language models. In *The Twelfth International Conference on Learning Representations*, 2024. 1

Nick Littlestone and Manfred K. Warmuth. The weighted majority algorithm. *Inf. Comput.*, 108(2): 212–261, 1994. 15

Stuart P. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28 (2):129–137, 1982. 1, 13

James MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pp. 281–297, 1967. 1, 13

H. Brendan McMahan. Follow-the-regularized-leader and mirror descent: Equivalence theorems and L1 regularization. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, AISTATS*, pp. 525–533, 2011. 15

Jiazhong Nie, Wojciech Kotłowski, and Manfred K. Warmuth. Online PCA with optimal regret. *Journal of Machine Learning Research*, 17(173):1–49, 2016. URL `http://jmlr.org/papers/v17/nie16a.html`. 14

Serge A. Plotkin, David B. Shmoys, and Éva Tardos. Fast approximation algorithms for fractional packing and covering problems. *Math. Oper. Res.*, 20(2):257–301, 1995. 15

David A Ross, Jongwoo Lim, Ruei-Sung Lin, and Ming-Hsuan Yang. Incremental learning for robust visual tracking. *International journal of computer vision*, 77(1-3):125–141, 2008. 13

Ruslan Salakhutdinov and Nathan Srebro. Collaborative filtering in a non-uniform world: Learning with the weighted trace norm. In *Advances in Neural Information Processing Systems*, volume 23, pp. 2056–2064, 2010. 14

Uri Shalit, Daphna Weinshall, and Gal Chechik. Online learning in the manifold of low-rank matrices. In *Advances in Neural Information Processing Systems*, volume 23, 2010. 15

Ohad Shamir and Shai Shalev-Shwartz. Collaborative filtering with the trace norm: Learning, bounding, and transducing. In *Proceedings of the 24th Annual Conference on Learning Theory*, volume 19 of *Proceedings of Machine Learning Research*, pp. 661–678, 2011. 14

Ohad Shamir and Shai Shalev-Shwartz. Matrix completion with the trace norm: Learning, bounding, and transducing. *Journal of Machine Learning Research*, 15(98):3401–3423, 2014. 14

Nathan Srebro and Tommi S. Jaakkola. Weighted low-rank approximations. In *Machine Learning, Proceedings of the Twentieth International Conference(ICML)*, pp. 720–727, 2003. 2, 14

Nathan Srebro and Adi Shraibman. Rank, trace-norm and max-norm. In *Proceedings of the 18th Annual Conference on Learning Theory*, pp. 545–560, 2005. 14

Xi Wang, Zhipeng Tu, Yiguang Hong, Yingyi Wu, and Guodong Shi. Online optimization over riemannian manifolds. *Journal of Machine Learning Research*, 24(84):1–67, 2023. 15

Manfred K. Warmuth and Dima Kuzmin. Randomized online PCA algorithms with regret bounds that are logarithmic in the dimension. *Journal of Machine Learning Research*, 9(75):2287–2320, 2008. 14

Bingyang Wu, Ruidong Zhu, Zili Zhang, Peng Sun, Xuanzhe Liu, and Xin Jin. {dLoRA}: Dynamically orchestrating requests and adapters for {LoRA}{LLM} serving. In *18th USENIX Symposium on Operating Systems Design and Implementation (OSDI 24)*, pp. 911–927, 2024. 1

Yuhui Xu, Lingxi Xie, Xiaotao Gu, Xin Chen, Heng Chang, Hengheng Zhang, Zhengsu Chen, Xiaopeng Zhang, and Qi Tian. Qa-lora: Quantization-aware low-rank adaptation of large language models. In *The Twelfth International Conference on Learning Representations, ICLR*, 2024. 1

A Related Work

Clustering and online k-means. The k-means objective is one of the most widely studied formulations of clustering, with origins in the work of MacQueen (1967) and the least-squares quantization procedure of Lloyd (1982). Given a dataset, the goal is to choose k centers minimizing the sum of squared distances from each point to its nearest center. Although the objective is simple and widely used, computing an optimal solution is computationally difficult, motivating a large literature on approximation algorithms, initialization schemes, and data summaries. The k-means++ seeding method of Arthur & Vassilvitskii (2007) gives a simple randomized initialization with logarithmic approximation guarantees, while local-search methods provide constant-factor approximation guarantees for the offline problem (Kanungo et al., 2004). Coresets have also played a central role in scalable clustering, allowing a small weighted summary of the data to preserve clustering costs for all candidate center sets (Har-Peled & Mazumdar, 2004).

The online version of k-means is more subtle because the learner must output centers before observing the next point and incurs loss immediately. This differs from classical streaming clustering, where the algorithm may process points sequentially but is only required to output a final solution after the stream has ended. The regret-based online k-means framework of Cohen-Addad et al. (2021) is closest to our setting: the learner maintains k candidate centers at each time step and competes with the best fixed k-means solution in hindsight. Their work shows that multiplicative weights over a discretized grid can achieve sublinear regret information-theoretically, while efficient exact regret minimization is obstructed by online-to-offline hardness. It also develops an incremental coreset, adaptive hierarchical region decomposition, and Mass Tree MWUA method for a $(1+\varepsilon)$-approximate regret benchmark. Our k-means contribution builds on this regret perspective but uses projected Voronoi assignments and ambient running means. At fixed ε, the projection dimension and expert-class size have no explicit dependence on d or T, although updates still involve d-dimensional data and tuning ε with T introduces an indirect horizon dependence.

Low-rank approximation. Low-rank approximation has a rich history in both linear algebra and machine learning, serving as a foundational technique for data compression, dimensionality reduction, and uncovering latent structure. The seminal result of Eckart & Young (1936) established that the optimal rank-k approximation of a matrix in the Frobenius norm is obtained via truncated singular value decomposition (SVD). Building on this, a wide range of classical algorithms have been developed for efficiently computing low-rank approximations of static datasets. Randomized SVD methods (Halko et al., 2011) exploit random projections to accelerate computation while preserving accuracy, and principal component analysis (PCA) (Jolliffe, 2002) remains a widely used method for identifying directions of maximal variance in high-dimensional data. These techniques are central to many modern applications, including collaborative filtering, natural language processing, and computer vision, where they reduce storage requirements, accelerate downstream computations, and improve interpretability.

The online or streaming setting, in which data arrives sequentially, introduces additional challenges. Unlike static datasets, where the full matrix is known prior, online low-rank approximation must maintain an accurate representation of the matrix as new rows or columns arrive, without recomputing the full SVD at each step. Incremental SVD updates (Brand, 2006) address this by algebraically modifying existing decompositions to account for additions or deletions of rows and columns, enabling efficient updates while retaining the low-rank structure. Similarly, online subspace identification from incomplete observations (Balzano et al., 2010) generalizes these ideas to sparse streaming matrices, which are common in recommendation systems and user-item interactions. These streaming methods address representation updates, whereas our online-learning protocol requires a subspace to be chosen before each observation and evaluates cumulative loss against a fixed hindsight comparator.

Online PCA extends these ideas further, enabling incremental dimensionality reduction in applications with continuously arriving high-dimensional data. For instance, Ross et al. (2008) apply online PCA to visual tracking, processing frames sequentially, projecting each onto the current subspace, and updating mean, covariance, and principal components. Subsequent work (Feng et al., 2013) addresses the limitations of classical online PCA methods by incorporating robustness against outliers and corrupted observations, which frequently arise in real-world streaming datasets. These ap-

proaches demonstrate the importance of maintaining adaptability and resilience in online low-rank models, particularly when the data distribution is non-stationary or contains noise.

Collectively, these contributions highlight both the theoretical foundations and practical algorithms for online low-rank approximation. They motivate the need for methods that balance computational efficiency with adaptive accuracy, especially in high-dimensional, streaming, and noisy environments. Our work studies this problem from a regret-minimization perspective, distinguishing exact rank-k comparison, $(1+\varepsilon)$ rank-k comparison, and optimization of a tractable fractional surrogate.

Online PCA and online subspace learning. The unweighted version of online low-rank approximation is closely related to online PCA, where the learner selects a low-dimensional subspace and suffers either compression loss or reconstruction gain on the next vector. This problem has been studied extensively, and several works obtain sublinear regret without discretizing the Grassmannian into a finite expert class. Warmuth & Kuzmin (2008) introduced randomized online PCA algorithms with regret bounds logarithmic in the ambient dimension, using matrix multiplicative-weights style updates over density matrices. Nie et al. (2016) later gave optimal regret guarantees for online PCA, analyzing gradient descent and exponentiated-gradient methods in the appropriate projection-matrix parameterization. For the rank-one case and related spectral objectives, Garber et al. (2015) studied online learning of eigenvectors and gave efficient regret-minimization algorithms. These results show that, in the ordinary unweighted PCA setting, one can often avoid an explicit finite net of subspaces by working directly with convex spectral relaxations.

Our work differs from this as the low-rank results focus on the weighted reconstruction loss, where the diagonal weights vary across coordinates and the corresponding offline problem is NP-hard. Thus, the main role of Grassmannian HRD is not to rederive known online PCA guarantees, but to provide an exact rank-k comparator framework for the weighted, nonconvex setting and to clarify why polynomial-time algorithms require a relaxed comparator.

Weighted low-rank approximation and convex relaxations. Weighted low-rank approximation generalizes classical low-rank approximation by assigning different importance weights to different entries of the matrix. Given a target matrix A and nonnegative weights W_{ij}, the objective is to find a low-rank matrix L minimizing

$$\sum_{i,j} W_{ij}(A_{ij} - L_{ij})^2.$$

This formulation captures heterogeneous confidence, missing entries, and coordinate-dependent noise, and appears naturally in collaborative filtering, matrix completion, computer vision, and structure-from-motion problems. Srebro & Jaakkola (2003) introduced weighted low-rank approximation as a general framework and proposed an EM-style algorithm for solving such problems in practice. In contrast with ordinary Frobenius-norm low-rank approximation, the weighted problem is computationally hard: Gillis & Glineur (2011) showed that computing an optimal weighted low-rank approximation is NP-hard, even for rank one and strictly positive weights, including hardness at a prescribed inverse-polynomial accuracy scale. Our online-to-offline reduction uses this accuracy scale to constrain sufficiently strong online guarantees against the standard WLRA comparator; it does not establish a barrier for every fixed constant ε.

A related line of work uses convex relaxations for low-rank learning and matrix completion, including trace-norm and spectral relaxations. These approaches replace an explicit rank constraint with a convex surrogate, enabling efficient optimization under appropriate assumptions. For example, trace-norm regularization has been used extensively in matrix completion and collaborative filtering (Srebro & Shraibman, 2005; Salakhutdinov & Srebro, 2010; Shamir & Shalev-Shwartz, 2011; 2014). Our Fantope relaxation follows the same broad philosophy, but the object being relaxed is the set of rank-k projection matrices rather than the set of low-rank matrices themselves. The resulting feasible set is the convex hull of rank-k projectors, namely the set of positive semidefinite matrices with eigenvalues in $[0,1]$ and trace k. This connects our polynomial-time online WLRA algorithm to the online PCA literature, while also making clear why the guarantee is against a fractional comparator rather than the exact Grassmannian optimum.

Grassmannian geometry and subspace methods. The Grassmann manifold $\mathrm{Gr}(k,d)$, the space of k-dimensional linear subspaces of $\mathbb{R}^d$, provides the natural geometric setting for low-rank approximation and subspace learning. Working directly on the Grassmannian eliminates basis dependence and enables invariant formulations of subspace tracking and approximation problems. The geome-

try of the Grassmannian is characterized by *principal angles* between subspaces (Björck & Golub, 1973), which quantify subspace similarity and induce natural distance metrics such as the chordal distance and geodesic distance. These geometric structures have been extensively studied in the context of optimization algorithms on matrix manifolds (Edelman et al., 1998; Absil et al., 2008), where gradient-based methods exploit the manifold structure for efficient computation. In machine learning and computer vision, Grassmannian methods have been applied to problems including subspace clustering, video analysis, and face recognition, where data naturally lies in or near low-dimensional subspaces. Our work uses bounded graph charts of the Grassmannian in two ways: to construct a fixed hierarchy of rank-k representatives for exact-comparator learning, and to define cells within which a rank-k learner moves for the $(1+\varepsilon)$ comparison.

Online optimization on manifolds. Another related direction is online optimization over Riemannian manifolds. For low-rank matrix manifolds, Shalit et al. (2010) proposed online learning algorithms based on Riemannian gradient steps and retractions, allowing low-rank structure to be maintained throughout the optimization process. More recently, Wang et al. (2023) developed online gradient and bandit algorithms over Riemannian manifolds, giving regret bounds for geodesically convex and strongly geodesically convex losses in appropriate manifold settings. These works are relevant because the Grassmannian is itself a Riemannian manifold, and they provide general tools for online optimization beyond Euclidean domains.

Our setting differs in the role played by the manifold. We do not assume that standard weighted reconstruction loss is geodesically convex over the Grassmannian or use Riemannian gradient descent to track a single subspace. For the exact-comparator result, we refine bounded graph charts to a fixed depth before the stream and run Mass Tree MWUA over their leaf representatives. For the $(1+\varepsilon)$ result, projected gradient descent runs on a convex surrogate within each fixed chart cell, while MWUA selects among cells using their true weighted reconstruction losses. The latter gradient updates take place in chart coordinates, and every iterate still represents a rank-k subspace.

Online learning with experts. Online learning with experts and related sequential decision-making problems have been extensively studied in the machine learning and theoretical computer science communities (Cesa-Bianchi & Lugosi, 2006). In this framework, an algorithm repeatedly selects actions or "experts" from a predefined set, receives feedback in the form of losses, and aims to minimize its cumulative loss relative to the best fixed expert in hindsight. Early work assumed the existence of a "perfect expert" whose predictions were always correct. In this idealized scenario, a folklore "halving" algorithm achieves at most $\log_2 n$ mistakes by retaining the set of experts that have not yet erred and predicting based on majority vote.

Recognizing that a perfect expert rarely exists in practice, Littlestone & Warmuth (1994) introduced the randomized weighted majority (RWM) algorithm. Here, predictions are sampled according to a probability distribution proportional to the experts' weights, which are updated multiplicatively after each round. This yields $O(\sqrt{T \log n})$ regret and is asymptotically optimal (Cover, 1967). MWUA, also called multiplicative weights in this setting, applies the same exponential-weighting idea to a finite class of experts. More generally, the Multiplicative Weights Update (MWU) framework extends these ideas to arbitrary loss functions and has found applications in boosting (e.g., AdaBoost (Freund & Schapire, 1997)), approximately solving zero-sum games (Freund & Schapire, 1999), and efficiently approximating linear and semi-definite programs (Clarkson, 1995; Plotkin et al., 1995; Garg & Könemann, 2007). Variants such as Follow the Perturbed Leader (FTPL) and Follow the Regularized Leader (FTRL) introduce random perturbations or regularization to improve stability and computational efficiency in structured problems (Kalai & Vempala, 2005; McMahan, 2011). For an in-depth survey of MWUA and its widespread applications in online optimization, game theory, and combinatorial learning, see Arora et al. (2012).

A particularly relevant extension is the work on online learning with low-rank experts by Hazan et al. (2016), which exploits hidden low-rank structure in the expert loss matrix. In this setting, although the nominal number of experts may be very large, the effective dimensionality of the loss matrix is small, allowing regret bounds to scale with the rank rather than the total number of experts. This perspective is related to our use of structured hypothesis classes, but the technical setting is different: our losses arise from geometric approximation problems, and the relevant hypothesis spaces are center tuples, Grassmannian subspaces, or convex spectral relaxations.

Variants of MWUA have also been developed to handle hierarchical or tree-structured expert sets. Notably, the Mass Tree MWUA (MTMW) framework of Cohen-Addad et al. (2021) introduces a hierarchical organization of experts, allowing multiplicative updates to be applied over adaptively revealed tree-structured classes. This hierarchical approach provides both computational advantages and refined control over exploration across different scales of the expert set.

The expert-based algorithms in this paper use these tools in different ways. Online k-means uses MWUA over a flat class of projected Voronoi experts, each of which tracks ambient running means. Exact-comparator WLRA uses MTMW over fixed Grassmannian leaf representatives. The $(1+\varepsilon)$ WLRA algorithm uses MWUA over chart cells containing moving projected-gradient learners. The reduced Fantope method takes a separate route, applying online gradient descent to a convex weighted projection surrogate.

B PRELIMINARIES

We collect the notation and online-learning tools used in the proofs. The two WLRA objectives in the paper are defined separately below: the Grassmannian algorithms use standard weighted reconstruction loss, while the Fantope algorithm uses a weighted projection surrogate.

Notation and online protocol. Write $[n] = \{1, \dots, n\}$. For a matrix A, let $\|A\|_F$ and $\|A\|_{\mathrm{op}}$ denote its Frobenius and operator norms. We write $A \succeq 0$ when A is positive semidefinite and use I_n for the $n \times n$ identity matrix.

On each round $t \in [T]$, the learner chooses a hypothesis h_t before observing the round's data and incurring loss $\ell_t(h_t)$. For a fixed comparator class $\mathcal{H}$, its regret is

$$\sum_{t=1}^{T} \ell_t(h_t) - \min_{h \in \mathcal{H}} \sum_{t=1}^{T} \ell_t(h).$$

Our randomized guarantees take expectation over the learner's choices and, in the k-means result, the random projection. The comparator class and even the loss being compared differ between the standard WLRA and Fantope results, so we specify each benchmark explicitly.

Online k-means. The learner chooses k centers $C_t = \{c_{t,1}, \dots, c_{t,k}\} \subseteq \mathbb{R}^d$ before observing $x_t \in \mathbb{R}^d$ and incurs

$$\ell_t(C_t) = \min_{c \in C_t} \|x_t - c\|_2^2.$$

For a best fixed center set $C^\star = \{c_1^\star, \dots, c_k^\star\}$ in hindsight, define

$$\mathcal{C}_{\mathrm{OPT}} := \sum_{t=1}^{T} \min_{j \in [k]} \|x_t - c_j^\star\|_2^2.$$

We assume $\|x_t\|_2 \le 1$ and $\|c_j^\star\|_2 \le 1$. The stream is chosen independently of the Gaussian projection used by the algorithm; in particular, $x_{1:T}$ and $C^\star$ can be fixed before that projection is sampled.

Marginal projection and ambient center tracking. The k-means algorithm samples a matrix $\Pi \in \mathbb{R}^{m \times d}$ with independent $\mathcal{N}(0, 1/m)$ entries. At any fixed round t, its analysis asks Π to preserve only the k distances from x_t to the centers in $C^\star$. A Gaussian Johnson–Lindenstrauss bound therefore gives

$$m = O\left(\varepsilon^{-2} \log \frac{k}{\varepsilon}\right)$$

without a union bound over all T rounds. This is a marginal guarantee: it does not assert that every distance is preserved simultaneously throughout the stream.

A projected grid tuple defines an expert's Voronoi assignments. Each expert separately maintains, for every assigned cluster, its count and running mean in $\mathbb{R}^d$. Those means are Follow-the-Leader (FTL) predictions for the points assigned to that cluster. Thus projected coordinates determine routing, while the predicted centers and their losses remain in the ambient space.

Standard online WLRA. Let $1 \le k < d$ and let $\mathrm{Gr}(k, d)$ be the set of k-dimensional subspaces of $\mathbb{R}^d$. We identify a subspace U with its orthogonal projector P_U and write

$$\mathcal{P}_k(d) = \{P \in \mathbb{R}^{d \times d} : P = P^\top,\ P^2 = P,\ \mathrm{tr}(P) = k\}.$$

For $P \in \mathcal{P}_k(d)$, let X_P be any full-column-rank matrix spanning its range. The learner chooses P_t before observing (x_t, W_t) and incurs the *standard weighted reconstruction loss*

$$\ell_t(P_t) = \min_{a \in \mathbb{R}^k} \left\| W_t^{1/2}(x_t - X_{P_t} a) \right\|_2^2.$$

This definition is independent of the choice of basis X_P. The exact rank-k benchmark is

$$C_{\mathrm{WLRA}} := \min_{P \in \mathcal{P}_k(d)} \sum_{t=1}^{T} \ell_t(P).$$

For the Grassmannian results, we assume $\|x_t\|_2 \le 1$ and

$$\mu I_d \preceq W_t \preceq W I_d, \qquad 0 < \mu \le W, \qquad \kappa := W/\mu.$$

The exact-comparator algorithm permits any weight matrix satisfying these spectral bounds. The locally convexified $(1+\varepsilon)$-comparator algorithm additionally requires W_t to be diagonal. In both cases, $0 \le \ell_t(P) \le W$.

Graph charts and Grassmannian distance. For a k-element set $J \subseteq [d]$, order the coordinates as (J, J^c). A subspace whose basis has an invertible minor in rows J can be represented by

$$X_J(Y) = \begin{pmatrix} I_k \\ Y \end{pmatrix}, \qquad Y \in \mathbb{R}^{(d-k)\times k},$$

with the coordinate ordering reversed when forming the ambient projector. Its projector is

$$P_J(Y) = X_J(Y)(I_k + Y^\top Y)^{-1} X_J(Y)^\top.$$

Choosing a largest-magnitude $k \times k$ row minor ensures that the entries of Y lie in $[-1, 1]$. Hence the $\binom{d}{k}$ bounded chart boxes $[-1,1]^{(d-k)\times k}$ cover $\mathrm{Gr}(k,d)$, although they may overlap. Each box has $k(d-k)$ coordinates. The Grassmannian algorithms partition these boxes into cubes: the exact-comparator algorithm plays fixed cube representatives, and the $(1+\varepsilon)$ algorithm moves within a cube.

For subspaces U, V, we use the chordal distance

$$d_G(U,V) := \frac{1}{\sqrt{2}} \|P_U - P_V\|_F = \left(\sum_{i=1}^{k} \sin^2 \theta_i \right)^{1/2},$$

where $\theta_1, \ldots, \theta_k$ are their principal angles. This distance connects chart resolution to approximation error between rank-k projectors.

Expert and online convex-learning bounds. For N experts with losses in $[0, \rho]$, MWUA, or equivalently multiplicative weights, gives expected loss at most

$$\min_{e \in [N]} \sum_{t=1}^{T} \ell_t(e) + O\left(\rho \sqrt{T \log N}\right).$$

We use this bound for projected Voronoi experts in k-means and to select among chart cubes in the locally convexified WLRA algorithm. An expert may maintain its own state, such as running cluster means or a within-cube learner; its prediction is determined before the current data are revealed.

The exact-comparator WLRA algorithm organizes fixed representatives as leaves of a tree and uses Mass Tree MWUA. Starting with unit mass at the root and splitting each node's mass among its children assigns a prior mass $M(p)$ to each leaf path p. Its comparison term scales as

$$O\left(\rho \sqrt{T \log \frac{1}{M(p)}}\right).$$

The hierarchy in that algorithm is refined to its chosen depth before the first round. The locally convexified algorithm instead uses projected gradient descent within each fixed cube. We also use

projected gradient descent over the Fantope below; for convex losses with bounded gradients on a bounded domain, its standard regret bound is sublinear in T.

Fantope surrogate and reduced comparator. The convex hull of $\mathcal{P}_k(d)$ is the Fantope

$$\mathcal{F}_k(d) = \{Q \in \mathbb{R}^{d\times d} : Q = Q^\top,\ 0 \preceq Q \preceq I_d,\ \mathrm{tr}(Q) = k\}.$$

Its elements are fractional projectors: their eigenvalues lie in $[0, 1]$ and sum to k. To obtain smaller updates, fix an anchor matrix $V \in \mathbb{R}^{d\times m}$ with $V^\top V = I_m$ and $k \le m \le d$. The algorithm chooses $Q \in \mathcal{F}_k(m)$ and lifts it to $VQV^\top \in \mathcal{F}_k(d)$.

The Fantope algorithm uses the *weighted projection surrogate*

$$\widetilde{\ell}_t(Q) := \left\|W_t^{1/2}(I_d - VQV^\top)x_t\right\|_2^2$$

and compares with the best fixed $Q \in \mathcal{F}_k(m)$ under this same loss. Its objective is convex in Q and admits an exact expression using m-dimensional sufficient statistics, developed in Appendix F.

Even when Q is an integral projector, this surrogate generally differs from standard WLRA for nonuniform W_t: standard WLRA minimizes over the reconstruction coefficients within the selected subspace, whereas the surrogate applies the orthogonal projector directly. They agree for scalar weights $W_t = w_t I_d$ and an integral projector. The Fantope guarantee therefore concerns both a different loss and a reduced fractional comparator class; it is not a regret bound against C_{WLRA}.

C Online k-Means via Marginal JL

We use the notation of Section 3. The aim of this section is to justify the analysis used in the proof of Theorem 3.1: the marginal Johnson–Lindenstrauss comparison, the ambient FTL tracking bound, and the final MWUA comparison.

The analysis can be viewed as the k-means analogue of the loss-decoupling argument used for the fantope relaxation. In the fantope setting, the ambient weighted loss is rewritten exactly as a reduced convex objective. Here, the projected anchors are used only for routing, while the representative centers are accumulated directly in $\mathbb{R}^d$. Thus the proof separates the low-dimensional assignment rule from the ambient representation error.

We first prove the marginal JL comparison used in Lemma C.1. The important point is that the projection is not required to preserve all distances over the full stream simultaneously. For each fixed time t, it only needs to preserve the k distances from x_t to the fixed offline centers.

Lemma C.1. *For each realization of Π, define $Y^\star = (y_1^\star, \dots, y_k^\star) \in S_m^k$ as follows. If $\|\Pi c_j^\star\|_2 \le 2$ for every $j \in [k]$, choose each $y_j^\star$ within distance ε of $\Pi c_j^\star$. Otherwise, set $Y^\star$ to any fixed element of S_m^k. Define*

$$i_t := \arg\min_{j\in[k]} \|x_t - c_j^\star\|_2^2, \qquad a_t := \arg\min_{j\in[k]} \|\Pi x_t - y_j^\star\|_2^2.$$

Then, taking expectation over the random projection Π,

$$\mathbb{E}_\Pi\left[\|x_t - c_{a_t}^\star\|_2^2\right] \le (1 + O(\varepsilon))\|x_t - c_{i_t}^\star\|_2^2 + O(\varepsilon).$$

Proof. Fix a time t. Let

$$i_t = \arg\min_{j\in[k]} \|x_t - c_j^\star\|_2^2, \qquad a_t = \arg\min_{j\in[k]} \|\Pi x_t - y_j^\star\|_2^2.$$

Let G be the event that $\|\Pi c_j^\star\|_2 \le 2$ for every $j \in [k]$. A Gaussian JL bound applied to the k fixed centers gives $\Pr(G^c) \le \varepsilon$: indeed, preservation of their squared norms within $(1 \pm \varepsilon)$ implies $\|\Pi c_j^\star\|_2 \le \sqrt{1+\varepsilon} < 2$.

Let D_t be the event that Π preserves the squared norms of $x_t - c_1^\star, \dots, x_t - c_k^\star$ within $(1 \pm \varepsilon)$. A Gaussian JL bound and a union bound over these k fixed vectors give $\Pr(D_t^c) \le \varepsilon$ for the stated choice of m. Set $E_t = G \cap D_t$. Then $\Pr(E_t^c) \le 2\varepsilon$. No union bound over the rounds is taken. Condition on E_t. Since a_t minimizes the projected distance to the grid anchors,

$$\|\Pi x_t - y_{a_t}^\star\|_2 \le \|\Pi x_t - y_{i_t}^\star\|_2.$$

Using the grid approximation $\|y_j^\star - \Pi c_j^\star\|_2 \le \varepsilon$ and the triangle inequality,

$$\begin{aligned}\|\Pi x_t - \Pi c_{a_t}^\star\|_2 &\le \|\Pi x_t - y_{a_t}^\star\|_2 + \varepsilon \\ &\le \|\Pi x_t - y_{i_t}^\star\|_2 + \varepsilon \\ &\le \|\Pi x_t - \Pi c_{i_t}^\star\|_2 + 2\varepsilon.\end{aligned}$$

Applying the JL inequalities on E_t gives

$$\sqrt{1-\varepsilon}\,\|x_t - c_{a_t}^\star\|_2 \le \sqrt{1+\varepsilon}\,\|x_t - c_{i_t}^\star\|_2 + 2\varepsilon.$$

Squaring both sides and using $\|x_t\|_2, \|c_j^\star\|_2 \le 1$, so that all relevant distances are at most 2, yields

$$\|x_t - c_{a_t}^\star\|_2^2 \le (1 + O(\varepsilon))\|x_t - c_{i_t}^\star\|_2^2 + O(\varepsilon).$$

On the complement E_t^c, the squared distance is at most 4, and $\Pr(E_t^c) \le 2\varepsilon$, contributing an additional $O(\varepsilon)$ term. Taking expectation over Π proves

$$\mathbb{E}_\Pi\big[\|x_t - c_{a_t}^\star\|_2^2\big] \le (1 + O(\varepsilon))\|x_t - c_{i_t}^\star\|_2^2 + O(\varepsilon).$$

□

We next justify the ambient tracking term. Once a projected expert fixes a routing rule, each cluster receives a subsequence of ambient points. The running mean on each such subsequence competes with the best fixed center for that subsequence up to a logarithmic penalty.

Lemma C.2. *Let $v_1, \dots, v_n \in B_d$, and let $\mu_i = \frac{1}{i-1}\sum_{r<i} v_r$ for $i \ge 2$, with $\mu_1 = 0$. Then*

$$\sum_{i=1}^n \|v_i - \mu_i\|_2^2 \le \min_{c\in\mathbb{R}^d} \sum_{i=1}^n \|v_i - c\|_2^2 + O(\log n).$$

Consequently, for any expert-induced partition into k clusters, the total FTL tracking penalty is $O(k \log T)$.

Proof. Let

$$E_i := \min_{c\in\mathbb{R}^d} \sum_{r=1}^i \|v_r - c\|_2^2 = \sum_{r=1}^i \|v_r - \bar{v}_i\|_2^2, \qquad \bar{v}_i := \frac{1}{i}\sum_{r=1}^i v_r.$$

For $i \ge 2$, the prediction μ_i is exactly the empirical mean of the previous $i-1$ points. The standard variance update identity gives

$$E_i - E_{i-1} = \frac{i-1}{i}\|v_i - \mu_i\|_2^2.$$

Thus

$$\|v_i - \mu_i\|_2^2 = \left(1 + \frac{1}{i-1}\right)(E_i - E_{i-1}).$$

Since $v_i, \mu_i \in B_d$, we have $\|v_i - \mu_i\|_2^2 \le 4$, and hence $E_i - E_{i-1} \le 4$. Summing over $i \ge 2$,

$$\begin{aligned}\sum_{i=2}^n \|v_i - \mu_i\|_2^2 &= \sum_{i=2}^n (E_i - E_{i-1}) + \sum_{i=2}^n \frac{E_i - E_{i-1}}{i-1} \\ &\le E_n + 4\sum_{i=2}^n \frac{1}{i-1} \\ &\le E_n + O(\log n).\end{aligned}$$

The first-round loss is bounded by $\|v_1 - \mu_1\|_2^2 \le 1$, so it is absorbed into the logarithmic term. Therefore,

$$\sum_{i=1}^n \|v_i - \mu_i\|_2^2 \le \min_{c\in\mathbb{R}^d} \sum_{i=1}^n \|v_i - c\|_2^2 + O(\log n).$$

Applying this bound separately to the subsequence assigned to each of the k clusters gives the total penalty $O(k \log T)$. □

Theorem 3.1. *Run Algorithm 1 with $\eta = \Theta(\sqrt{\log|\mathcal{E}|/T})$. Then*

$$\mathbb{E}_{\Pi,\mathrm{alg}}\left[\sum_{t=1}^{T}\ell_t(C_t)\right] \le (1+O(\varepsilon))\mathcal{C}_{\mathrm{OPT}} + O(\varepsilon T) + O\left(\sqrt{T\log|\mathcal{E}|}\right) + O(k\log T).$$

Moreover, $\log|\mathcal{E}| = O\left(km\log\frac{\sqrt{m}}{\varepsilon}\right)$ and $m = O\left(\varepsilon^{-2}\log\frac{k}{\varepsilon}\right)$, so the expert complexity is independent of both the ambient dimension d and the time horizon T.

Proof of Theorem 3.1. Fix the projection Π. For the surrogate losses

$$\widetilde{\ell}_t(Y) = \|x_t - \mu^{Y}_{t,a_t^Y}\|_2^2,$$

the standard MWUA bound gives

$$\mathbb{E}_{\mathrm{alg}}\left[\sum_{t=1}^{T}\widetilde{\ell}_t(Y_t)\right] \le \sum_{t=1}^{T}\widetilde{\ell}_t(Y^\star) + O\left(\sqrt{T\log|\mathcal{E}|}\right),$$

where $Y^\star$ is the projection-dependent grid expert defined in Lemma C.1. The learner's actual loss is no larger than the surrogate loss of the sampled expert, since

$$\ell_t(C_t(Y_t)) = \min_{j\in[k]}\|x_t - \mu^{Y_t}_{t,j}\|_2^2 \le \|x_t - \mu^{Y_t}_{t,a_t^{Y_t}}\|_2^2 = \widetilde{\ell}_t(Y_t).$$

Thus it remains to bound the cumulative surrogate loss of $Y^\star$.

Let

$$a_t := \arg\min_{j\in[k]}\|\Pi x_t - y_j^\star\|_2^2$$

be the projected assignment induced by $Y^\star$. This assignment partitions the stream into k subsequences. Applying Lemma C.2 to each induced subsequence gives

$$\sum_{t=1}^{T}\widetilde{\ell}_t(Y^\star) \le \sum_{j=1}^{k}\min_{c\in\mathbb{R}^d}\sum_{t:a_t=j}\|x_t - c\|_2^2 + O(k\log T).$$

For each cluster j, the offline center $c_j^\star$ is a feasible fixed center for the corresponding induced subsequence. Therefore,

$$\sum_{t=1}^{T}\widetilde{\ell}_t(Y^\star) \le \sum_{t=1}^{T}\|x_t - c^\star_{a_t}\|_2^2 + O(k\log T).$$

Taking expectation over Π and applying Lemma C.1 at each time step,

$$\begin{aligned}\mathbb{E}_\Pi\left[\sum_{t=1}^{T}\|x_t - c^\star_{a_t}\|_2^2\right] &= \sum_{t=1}^{T}\mathbb{E}_\Pi\left[\|x_t - c^\star_{a_t}\|_2^2\right]\\ &\le (1+O(\varepsilon))\sum_{t=1}^{T}\min_{j\in[k]}\|x_t - c_j^\star\|_2^2 + O(\varepsilon T)\\ &= (1+O(\varepsilon))\mathrm{OPT} + O(\varepsilon T).\end{aligned}$$

Combining the MWUA comparison, the surrogate-to-actual loss inequality, and the ambient tracking bound proves

$$\mathbb{E}_{\Pi,\mathrm{alg}}\left[\sum_{t=1}^{T}\ell_t(C_t)\right] \le (1+O(\varepsilon))\mathrm{OPT} + O(\varepsilon T) + O\left(\sqrt{T\log|\mathcal{E}|}\right) + O(k\log T).$$

It remains only to bound the size of the expert class. The grid satisfies

$$|S_m| \le \left(\frac{C\sqrt{m}}{\varepsilon}\right)^m$$

for a universal constant C. Since an expert is a k-tuple of projected grid points,

$$|\mathcal{E}| \le |S_m|^k, \qquad \log|\mathcal{E}| = O\left(km \log \frac{\sqrt{m}}{\varepsilon}\right).$$

Substituting

$$m = O\left(\varepsilon^{-2} \log \frac{k}{\varepsilon}\right)$$

shows that the expert complexity has no dependence on T or on the ambient dimension d. □

The preceding guarantee should be interpreted as removing the horizon-dependent and ambient-dimension-dependent discretization barrier from the regret analysis. The projected expert class still has size exponential in k and ε^{-1}, but its logarithm is independent of both T and d. Thus the result plays the same conceptual role for online k-means that the reduced fantope formulation plays for WLRA: it separates the statistical comparison from the original ambient geometry. The difference is that the fantope relaxation replaces the discrete expert class by a convex feasible region, while the projected k-means result remains an experts-based discretization.

D Grassmannian HRD for Exact-Comparator Online WLRA

We prove Theorem 4.1 for the standard WLRA loss. The argument has three parts. First, bounded graph charts give a finite collection of rank-k representatives, including one close to the cumulative optimum $P^\star$. Second, the cumulative loss at that representative differs from C_{WLRA} by $O(W\sqrt{T})$. Third, MTMW competes with this fixed representative using its path mass. We give the chart construction and the algorithm before proving the two loss bounds and combining them.

D.1 Chart decomposition of the Grassmannian

To partition the Grassmannian into small regions, we need coordinates for each subspace. We obtain them by choosing k reference rows of a basis matrix and writing the remaining rows as a matrix Y. The first lemma shows that every subspace has such a representation with entries of Y in $[-1, 1]$. The second shows that nearby coordinate matrices give nearby subspaces, measured by the distance between their orthogonal projectors.

Lemma D.1 (Bounded graph charts)**.** *For each k-element subset $J \subseteq [d]$, permute the rows in J to the top and define, for $Y \in [-1,1]^{(d-k)\times k}$,*

$$X_J(Y) = \begin{pmatrix} I_k \\ Y \end{pmatrix}, \qquad P_J(Y) = X_J(Y)\big(I_k + Y^\top Y\big)^{-1} X_J(Y)^\top,$$

with the row permutation reversed in the resulting ambient projector. Then:

(1) the $\binom{d}{k}$ charts cover $\mathcal{P}_k(d)$;

(2) if $P = P_J(Y)$, then $P_{J,J}$ is invertible and

$$Y = P_{J^c,J}(P_{J,J})^{-1};$$

(3) every $P_J(Y)$ has rank exactly k, and is rational whenever Y is rational.

Proof. Take a full-column-rank basis matrix of a k-dimensional subspace. Choose a $k \times k$ row minor with maximum absolute determinant and let J be its row indices. This minor is nonzero. Multiplying the basis on the right by its inverse makes the selected rows equal to I_k without changing the column space. By Cramer's rule, each remaining entry is, up to sign, the ratio of another $k \times k$ minor to the selected one. Its magnitude is therefore at most 1, proving the chart cover.

The blocks of $P_J(Y)$ are

$$P_{J,J} = (I_k + Y^\top Y)^{-1}, \qquad P_{J^c,J} = Y(I_k + Y^\top Y)^{-1}.$$

The first block is invertible, and multiplying the second by its inverse recovers Y.

Finally, the columns of $X_J(Y)$ contain an identity minor, so they are linearly independent and $P_J(Y)$ has rank k. If Y is rational, then $I_k + Y^\top Y$ and its inverse are rational, so $P_J(Y)$ is rational. □

Lemma D.2 (Stability within a graph chart)**.** *For every k-element subset $J \subseteq [d]$ and all $Y, Z \in [-1,1]^{(d-k)\times k}$,*

$$\|P_J(Y) - P_J(Z)\|_F \le \sqrt{2}\,\|Y - Z\|_F.$$

Proof. Write $X = X_J(Y)$, $X' = X_J(Z)$, and $U = X(X^\top X)^{-1/2}$. Then $U^\top U = I_k$ and $UU^\top = P_J(Y)$. Since

$$X^\top X = I_k + Y^\top Y \succeq I_k,$$

we have $\|(X^\top X)^{-1/2}\|_{\mathrm{op}} \le 1$. Also, $(I_d - P_J(Z))X' = 0$, so

$$\begin{aligned}\|(I_d - P_J(Z))U\|_F &= \|(I_d - P_J(Z))(X - X')(X^\top X)^{-1/2}\|_F \\ &\le \|X - X'\|_F = \|Y - Z\|_F.\end{aligned}$$

For any two rank-k projectors $P = UU^\top$ and Q,

$$\|P - Q\|_F^2 = 2k - 2\,\mathrm{tr}(U^\top Q U) = 2\|(I_d - Q)U\|_F^2.$$

Taking $Q = P_J(Z)$ proves the claim. □

Regions and representatives. The chart boxes can overlap, so we first assign each projector to one chart. Fix an ordering of the k-element subsets of $[d]$. A chart J is eligible for $P \in \mathcal{P}_k(d)$ if $P_{J,J}$ is invertible and

$$Y_J(P) := P_{J^c,J}(P_{J,J})^{-1} \in [-1,1]^{(d-k)\times k}.$$

Every projector has an eligible chart by Lemma D.1(i),(ii). We let $J(P)$ be the first eligible chart in the fixed ordering.

We now divide each chart box into dyadic cubes, viewing a coordinate matrix as a vector of its $k(d-k)$ entries. The tree has a virtual root with one child for each chart. At depth $h \ge 0$ below a chart, its cubes have side length 2^{1-h}. We use a fixed half-open endpoint convention, including the endpoint 1, so that each coordinate matrix belongs to exactly one cube at each depth.

A chart J and cube c define the region

$$R_{J,c} := \{P \in \mathcal{P}_k(d) : J(P) = J,\ Y_J(P) \in c\}.$$

Its representative is $P_{J,c} := P_J(\bar{Y}_c)$, where $\bar{Y}_c$ is the center of c, reshaped as a $(d-k)\times k$ matrix. Each projector belongs to exactly one region at every depth. Refinement divides a cube into $2^{k(d-k)}$ children. Some regions are empty, but we retain these to keep the tree's branching factor uniform.

The representative $P_{J,c}$ need not itself belong to $R_{J,c}$: the projector at the cube center might be assigned to an earlier eligible chart. We only use $P_{J,c}$ to approximate projectors that belong to $R_{J,c}$.

Lemma D.3 (Partition and radius)**.** *For every depth h, the regions $R_{J,c}$ partition the space $\mathcal{P}_k(d)$ with every child region being contained in its parent. Additionally, for every $P \in R_{J,c}$ with c at depth h,*

$$\|P - P_{J,c}\|_F \le \sqrt{2k(d-k)}\,2^{-h}, \qquad d_G(P, P_{J,c}) \le \sqrt{k(d-k)}\,2^{-h}.$$

Proof. The fixed chart ordering assigns each projector P a unique chart $J(P)$. Within that chart, the half-open cubes at depth h partition the coordinate box, so $Y_{J(P)}(P)$ belongs to exactly one cube. Thus P belongs to exactly one region at depth h. Each child cube lies inside its parent cube, so the corresponding child region lies inside its parent region.

Now let $P \in R_{J,c}$, where c has depth h. The side length of c is 2^{1-h}, so each entry of $Y_J(P)$ differs from the corresponding entry of the center $\bar{Y}_c$ by at most 2^{-h}. There are $k(d-k)$ entries, giving

$$\|Y_J(P) - \bar{Y}_c\|_F \le \sqrt{k(d-k)}\,2^{-h}.$$

Since $P = P_J(Y_J(P))$ and $P_{J,c} = P_J(\bar{Y}_c)$, Lemma D.2 gives

$$\|P - P_{J,c}\|_F \le \sqrt{2k(d-k)}\, 2^{-h}.$$

Finally, $d_G(P,Q) = \|P - Q\|_F/\sqrt{2}$ for rank-k projectors, which gives the chordal-distance bound. □

Depth and learning rate. To make the quadratic comparator error of Lemma D.5 at most $\frac{3\pi^2}{2} W\sqrt{T}$, we use the following depth and number of leaves

$$L = \left\lceil \log_2\!\left(\kappa \sqrt{2k(d-k)}\, T^{1/4}\right)\right\rceil, \qquad N_{\mathrm{ex}} = \binom{d}{k} 2^{k(d-k)L}.$$

All charts are built to this depth before the first round. For MTMW set $K = \log N_{\mathrm{ex}}$ and $\eta = \sqrt{8K/T}$. These are the parameters of Algorithm 4; the next corollary verifies the claimed resolution and leaf count.

Corollary D.4 (Depth and number of regions)**.** *Every leaf at depth L satisfies*

$$\|P - P_{J,c}\|_F \le \delta_L := \sqrt{2k(d-k)}\, 2^{-L} \le \kappa^{-1} T^{-1/4}$$

for all P in its region. The number of leaves is

$$N_{\mathrm{ex}} = \binom{d}{k} 2^{k(d-k)L} \le \binom{d}{k} \left(2\kappa \sqrt{2k(d-k)}\, T^{1/4}\right)^{k(d-k)},$$

and the whole tree has at most $2N + 1$ nodes.

Proof. By the choice of L, $2^L \ge \kappa\sqrt{2k(d-k)}\, T^{1/4}$, which gives $\delta_L \le \kappa^{-1} T^{-1/4}$ by Lemma D.3. Since L is the smallest such integer and $\kappa\sqrt{2k(d-k)}\, T^{1/4} > 1$, also $2^L < 2\kappa\sqrt{2k(d-k)}\, T^{1/4}$, which gives the bound on N_{ex}. Each internal chart node has $2^{k(d-k)} \ge 2$ children, so each chart subtree has at most twice as many nodes as leaves; adding the virtual root gives the final bound. □

Algorithm 3 Grassmannian Chart Decomposition

Require: Dimensions d, k, depth L
Ensure: Leaves $\mathcal{R}$ with representatives and masses
  $\mathcal{R} \leftarrow \emptyset$
  for each k-subset $J \subseteq [d]$, in the fixed order **do**
    for each dyadic cube $c \subseteq [-1,1]^{k(d-k)}$ of side 2^{1-L} **do**
      Add $R_{J,c}$ with basis $X_J(\bar{Y}_c)$, projector $P_J(\bar{Y}_c)$, and mass $\binom{d}{k}^{-1} 2^{-k(d-k)L}$ to $\mathcal{R}$
  return $\mathcal{R}$

Algorithm 4 Grassmannian HRD Regret Minimization

Require: Horizon T, rank k, dimension d, bounds $0 < \mu \le W$
Ensure: Sequence of rank-k projectors $P_1, \dots, P_T$
  Set L, N_{ex}, K, and η as defined above
  Build the leaves $\mathcal{R}$ with Algorithm 3; set $w(R) \leftarrow M(R)$ for each $R \in \mathcal{R}$
  for $t \leftarrow 1$ to T **do**
    Sample R_t with probability $w(R_t)/\sum_R w(R)$ and play $P_t = P_{R_t}$
    Receive (x_t, W_t) and incur loss $\ell_t(P_t)$
    for each $R \in \mathcal{R}$ **do**
      $w(R) \leftarrow w(R) \exp\!\big(-\eta\, \ell_t(P_R)/W\big)$

Remark 2 (Why the depth is fixed in advance)**.** *The quadratic bound of Lemma D.5 compares the cumulative loss at $P^\star$ with the cumulative loss at one* fixed *representative. It does not transfer to a time-varying sequence of representatives chosen by a loss-dependent refinement rule: the first-order*

errors on different intervals need not cancel. We retain the nested tree construction because it makes the geometry of the discretization explicit and may provide a starting point for future adaptive refinements. We therefore refine every chart to depth L before round 1. The general MTMW statement in Appendix D.3 also permits adaptive refinement, but the exact-comparator proof here uses this fixed tree.

D.2 Loss stability and quadratic approximation

For an orthonormal basis U of a subspace, write

$$\ell_t(U) := \min_{a\in\mathbb{R}^k} \|W_t^{1/2}(x_t - Ua)\|_2^2.$$

This notation is basis-invariant.

The exact-comparator analysis uses a second-order estimate for the whole horizon, rather than a pointwise estimate for each round. We first prove that estimate and then compare it with the first-order alternative.

Lemma D.5 (Quadratic approximation at the cumulative optimum)**.** *Let $F(P) = \sum_{t=1}^T \ell_t(P)$ and let $P^\star \in \arg\min_{P\in\mathcal{P}_k(d)} F(P)$. Then every $P \in \mathcal{P}_k(d)$ satisfies*

$$F(P) - F(P^\star) \le \frac{3\pi^2}{2} W\kappa^2 T \|P - P^\star\|_F^2.$$

Proof. We connect $P^\star$ to P by a Grassmannian geodesic and bound the second derivative of the loss along it. Let $U(0)$ and $U(1)$ be principal-vector orthonormal bases for their ranges. Along the principal-angle geodesic, the ith basis vector rotates through angle $s\theta_i$. With $\theta_{\max} := \max_i \theta_i$, this gives

$$\|U'(s)\|_{\mathrm{op}} \le \theta_{\max}, \qquad \|U''(s)\|_{\mathrm{op}} \le \theta_{\max}^2.$$

Fix one round and abbreviate W_t by $\widetilde{W}$. The normal matrix $G(s)$ and right-hand side $b(s)$ describe weighted least squares in the moving basis; $z(s)$ is the minimizing coefficient vector. Define

$$G(s) := U(s)^\top \widetilde{W} U(s), \qquad b(s) := U(s)^\top \widetilde{W} x_t, \qquad z(s) := G(s)^{-1} b(s).$$

Then

$$\ell_t(U(s)) = x_t^\top \widetilde{W} x_t - b(s)^\top z(s).$$

The bound $G(s) \succeq \mu I_k$ prevents this least squares problem from becoming singular. Because the fitted vector is a weighted orthogonal projection, $\|z(s)\|_2 \le \sqrt{\kappa}$. Differentiating the normal matrix and right-hand side along the geodesic gives

$$\|G'(s)\|_{\mathrm{op}} \le 2W\theta_{\max}, \quad \|G''(s)\|_{\mathrm{op}} \le 4W\theta_{\max}^2,$$
$$\|b'(s)\|_2 \le W\theta_{\max}, \qquad \|b''(s)\|_2 \le W\theta_{\max}^2,$$

Differentiating $G(s)z(s) = b(s)$ then gives

$$z'(s) = G(s)^{-1}\big(b'(s) - G'(s)z(s)\big), \qquad \|z'(s)\|_2 \le \kappa\theta_{\max}(1 + 2\sqrt{\kappa}).$$

The loss is a constant minus $q(s) := b(s)^\top z(s)$. Differentiating this quadratic term, using $Gz = b$ to collect the terms involving z'', yields

$$q''(s) = 2b''(s)^\top z(s) + 2b'(s)^\top z'(s) - 2z'(s)^\top G'(s) z(s) - z(s)^\top G''(s) z(s).$$

Each displayed term is now controlled by the bounds for G, b, z and their derivatives. Since $\kappa \ge 1$, they imply

$$|q''(s)| \le W\theta_{\max}^2\big(2\sqrt{\kappa} + 6\kappa + 8\kappa^{3/2} + 8\kappa^2\big) \le 24W\kappa^2\theta_{\max}^2.$$

Thus $|\frac{d^2}{ds^2}\ell_t(U(s))| \le 24W\kappa^2\theta_{\max}^2$.

This is a per-round curvature bound; stationarity enters only after the rounds have been summed. Let $g(s) := F(P_{U(s)})$. The geodesic is feasible for all real s, and $P^\star$ minimizes F, so $g'(0) = 0$. Taylor's theorem with integral remainder therefore gives

$$F(P) - F(P^\star) = \int_0^1 (1-s) g''(s)\, ds \le 12W\kappa^2 T\theta_{\max}^2.$$

Finally, convert the maximum principal angle into Frobenius distance between the two projectors:

$$\|P - P^\star\|_F^2 = 2\sum_i \sin^2\theta_i, \qquad \theta_{\max}^2 \le \frac{\pi^2}{8}\|P - P^\star\|_F^2.$$

Substituting $\theta_{\max}^2 \le (\pi^2/8)\|P - P^\star\|_F^2$ into the preceding Taylor bound proves the claim. □

Corollary D.6 (Comparator path discretization error)**.** *Let $P_{p^\star}$ be the representative of the depth-L leaf containing $P^\star$. Then*

$$\sum_{t=1}^{T} \ell_t(P_{p^\star}) \le C_{\mathrm{WLRA}} + \frac{3\pi^2}{2} W\sqrt{T}.$$

Proof. By Corollary D.4, $\|P_{p^\star} - P^\star\|_F \le \kappa^{-1}T^{-1/4}$. Apply Lemma D.5: the factor $W\kappa^2 T$ multiplies a squared distance of at most $\kappa^{-2}T^{-1/2}$, leaving $W\sqrt{T}$ and the stated constant. □

Remark 3 (Why positive weights matter)**.** *The assumption $\mu I_d \preceq W_t$ gives the uniform bound*

$$\left\|(U^\top W_t U)^{-1}\right\|_{\mathrm{op}} \le \frac{1}{\mu}.$$

With zero weights the standard WLRA loss can fail to be continuous on the Grassmannian. For example, take $W_{\mathrm{ex}} = \mathrm{diag}(1, 0)$, $x = e_1$, and $k = 1$. Every line $U_\theta = \mathrm{span}((\cos\theta, \sin\theta))$ with $\cos\theta \neq 0$ has loss zero, whereas $U_{\pi/2}$ has loss one. Thus a uniform Grassmannian net cannot approximate the loss without a quantitative positive lower bound.

D.3 Mass Tree MWUA

The geometry controls the cost of the comparator leaf. To control the learner's additional loss, we assign that leaf a prior mass and apply a tree version of multiplicative weights. For each node v, define $M(\mathrm{root}) = 1$ and $M(v) = M(\mathrm{par}(v))/\deg(\mathrm{par}(v))$. The mass of a path is the mass of its terminal node. The mass identity below ensures that the leaf priors sum to one. We state MTMW for trees that may be refined between rounds: splitting an active leaf also splits its current weight equally. Our fixed-depth algorithm is the special case with no subsequent splits.

Lemma D.7. *Let v be a node and $\tilde{\mathcal{T}}$ a finite subtree rooted at v that contains all children of each of its internal nodes. If $\tilde{V}$ is its leaf set, then $M(v) = \sum_{v' \in \tilde{V}} M(v')$.*

Proof. By induction on the height of $\tilde{\mathcal{T}}$. If $\tilde{\mathcal{T}} = \{v\}$, then $\tilde{V} = \{v\}$. Otherwise let $\mathcal{U}$ be the children of v. By the induction hypothesis applied to the subtree below each $u \in \mathcal{U}$,

$$\sum_{v' \in \tilde{V}} M(v') = \sum_{u \in \mathcal{U}} M(u) = \sum_{u \in \mathcal{U}} \frac{M(v)}{|\mathcal{U}|} = M(v).$$ □

Theorem D.8. *Consider a tree whose active leaves are the experts, initialized with weights equal to their masses, possibly refined adaptively between rounds with weights split equally among children. Suppose the losses $m_t(R)$ of the active representatives lie in $[0, 1]$, and update $w(R) \leftarrow w(R)e^{-\eta m_t(R)}$ with $\eta > 0$. For a root-to-leaf path p, let $m_t(p)$ be the loss of its active node on round t. Then the mixture losses $\hat{m}_t = \sum_R w(R)m_t(R)/\sum_R w(R)$ satisfy*

$$\sum_{t=1}^{T} \hat{m}_t - \sum_{t=1}^{T} m_t(p) \le \frac{\log(1/M(p))}{\eta} + \frac{\eta T}{8}.$$

For $K \ge \log(1/M(p))$ and $\eta = \sqrt{8K/T}$, the right-hand side is at most $\sqrt{TK/2}$. For an oblivious input sequence, the same bound holds for the expected loss of a leaf sampled with probability proportional to its weight. Each round takes time linear in the number of active leaves plus the cost of evaluating their losses.

Proof. The proof compares an upper bound on the total active weight with a lower bound from any one comparator path. Let Ω_t be that total before round t; by Lemma D.7, $\Omega_1 = 1$. A split preserves it because the children's weights sum to their parent's. Applying Hoeffding's lemma to a leaf drawn from the normalized weights gives

$$\Omega_{t+1} = \Omega_t \, \mathbb{E}_R\big[e^{-\eta m_t(R)}\big] \le \Omega_t \exp\Big(-\eta \hat{m}_t + \frac{\eta^2}{8}\Big), \qquad \text{so} \qquad \log \Omega_{T+1} \le -\eta \sum_t \hat{m}_t + \frac{\eta^2 T}{8}.$$

For the lower bound, follow the active ancestors of the terminal node of p. Each split multiplies that branch's weight by the reciprocal of the parent's degree, giving its path mass $M(p)$ overall. Each multiplicative update on the branch contributes $e^{-\eta m_t(p)}$. Because this branch's final weight is part of the total, we obtain

$$M(p) \exp\left(-\eta \sum_t m_t(p)\right) \le \Omega_{T+1}.$$

Taking logarithms of the lower bound and combining it with the upper bound on $\log \Omega_{T+1}$ gives the claimed regret inequality, even when refinements depend on past inputs. The choice of η gives $K/\eta + \eta T/8 = \sqrt{TK/2}$. For an oblivious sequence, the weights and refinements do not depend on the learner's samples, so the expected sampled loss on each round equals $\hat{m}_t$. Normalizing, sampling, and updating are linear scans over the active leaves. □

Corollary D.9. *For standard online WLRA with $\|x_t\|_2 \le 1$ and $\mu I_d \preceq W_t \preceq W I_d$, running MTMW on the normalized losses ℓ_t/W with $\eta = \sqrt{8K/T}$ gives, for any path p with $\log(1/M(p)) \le K$,*

$$\mathbb{E} \sum_{t=1}^{T} \ell_t(P_t) \le \sum_{t=1}^{T} \ell_t(P_{p,t}) + W\sqrt{TK/2},$$

where $P_{p,t}$ is the representative of the active node of p on round t.

Proof. Since $0 \le \ell_t \le W$, the normalized losses lie in $[0, 1]$. Apply Theorem D.8 and multiply by W. □

D.4 Putting the components together

Lemma D.10 (Comparator path mass)**.** *For charts refined to depth L, every root-to-leaf path p has*

$$\log \frac{1}{M(p)} = \log \binom{d}{k} + k(d-k)L \log 2.$$

Proof. Every root-to-leaf path first selects one of the $\binom{d}{k}$ charts and then makes L choices among $2^{k(d-k)}$ children. Its prior mass is

$$M(p) = \binom{d}{k}^{-1} 2^{-k(d-k)L}.$$

□

The preceding mass calculation gives $K = \log N_{\mathrm{ex}}$ for the comparator leaf. Its loss is bounded by Corollary D.6; combining the two estimates proves the main theorem.

Theorem 4.1. *Under the preceding assumptions, Algorithm 4 satisfies*

$$\mathbb{E} \sum_{t=1}^{T} \ell_t(P_t) - C_{\mathrm{WLRA}} \le O\Big(W\sqrt{k(d-k)T \log(\kappa k(d-k)T)}\Big).$$

In the exact-real arithmetic model, with scalar exponentiation and finite-distribution sampling counted as primitive operations, its total running time is $O\Big(T(4\kappa\sqrt{2k(d-k)}\,T^{1/4})^{k(d-k)} \operatorname{poly}(d)\Big)$.

Proof. All leaves are at depth L, so $\log(1/M(p^\star)) = \log N_{\mathrm{ex}} = K$. The tree is fixed before the first round, and hence $p^\star$ uses one representative throughout the horizon. By Corollaries D.6 and D.9,

$$\begin{aligned}\mathbb{E}\sum_{t=1}^{T}\ell_t(P_t) &\le \sum_{t=1}^{T}\ell_t(P_{p^\star}) + W\sqrt{TK/2}\\ &\le C_{\mathrm{WLRA}} + \frac{3\pi^2}{2}W\sqrt{T} + W\sqrt{TK/2}.\end{aligned}$$

Moreover, Corollary D.4 gives

$$K \le \log\binom{d}{k} + k(d-k)\log\Big(2\kappa\sqrt{2k(d-k)}\,T^{1/4}\Big).$$

For $1 \le k < d$, the first term is $O(k(d-k)\log(k(d-k)))$. Consequently,

$$K = O\big(k(d-k)\log\big(\kappa k(d-k)T\big)\big),$$

which proves the regret bound.

For the running time, the algorithm evaluates all N_{ex} representatives on every round. Store both the chart basis $X_R = X_J(\bar{Y}_c)$ and its projector for each representative. For any positive definite W_t under the stated spectral bounds, evaluating one expert uses

$$G_R = X_R^\top W_t X_R, \qquad c_R = X_R^\top W_t x_t, \qquad \ell_t(P_R) = x_t^\top W_t x_t - c_R^\top G_R^{-1} c_R,$$

and costs $O(d^2k + dk^2 + k^3)$ arithmetic operations for a general dense positive definite W_t (or $O(dk^2 + k^3)$ when W_t is diagonal). Sampling and updating take linear time in the N_{ex} leaves. Thus the total cost is $O(TN_{\mathrm{ex}}\,\mathrm{poly}(d))$. Since $1 \le k < d$, we have $\binom{d}{k} \le 2^{d-1} \le 2^{k(d-k)}$. Hence from the bound in Corollary D.4,

$$N_{\mathrm{ex}} \le \big(4\kappa\sqrt{2k(d-k)}\,T^{1/4}\big)^{k(d-k)},$$

which gives the stated running time. □

E Locally Convexified Grassmannian Algorithm

We now prove Theorem 4.2. The exact-comparator algorithm plays a fixed projector for each chart cube, whereas here each cube instead runs an online learner that can move within it. We first express the standard WLRA loss in chart coordinates. On a small cube, freezing one matrix in that expression gives a convex loss that approximates the true loss multiplicatively. We then analyze the learner in one cube and use MWUA to choose among cubes. Throughout, W_t is diagonal with $\mu I_d \preceq W_t \preceq WI_d$, $\kappa = W/\mu$, and $0 < \varepsilon \le 1$. We use the same chart notation of Appendix D.

E.1 The loss in chart coordinates

By Lemma D.1, every rank-k projector has a representation $P_J(Y)$ with $Y \in [-1,1]^{(d-k)\times k}$. We will also use the coordinate bound

$$\|Y\|_{\mathrm{op}} \le \|Y\|_F \le \sqrt{k(d-k)}, \qquad Y \in [-1,1]^{(d-k)\times k}.$$

The last inequality follows by counting the $k(d-k)$ entries of Y.

The next formula removes the minimization over fitting coefficients from the standard WLRA loss. It is the reason that freezing a matrix will be useful.

Lemma E.1 (Chart formula)**.** *For a chart J, a coordinate matrix Y, and a round t, define*

$$r_{t,J}(Y) := x_{t,J^c} - Yx_{t,J}, \qquad S_{t,J}(Y) := W_{t,J^c}^{-1} + YW_{t,J}^{-1}Y^\top.$$

Then

$$\ell_t(P_J(Y)) = r_{t,J}(Y)^\top S_{t,J}(Y)^{-1} r_{t,J}(Y).$$

The loss and $S_{t,J}(Y)^{-1}r_{t,J}(Y)$ can each be computed in $O(dk^2 + k^3)$ arithmetic operations.

Proof. Fix t and J, and abbreviate $A = W_{t,J}$, $B = W_{t,J^c}$, and $r = r_{t,J}(Y)$. In the basis $X_J(Y)$, a fitted vector has coordinates (a, Ya). Set $e = a - x_{t,J}$. Its residual has blocks $-e$ and $r - Ye$, respectively, so

$$\ell_t(P_J(Y)) = \min_{e \in \mathbb{R}^k} \{e^\top A e + (r - Ye)^\top B (r - Ye)\}.$$

The minimizing coefficient is $e^\star = (A + Y^\top B Y)^{-1} Y^\top B r$. Substitution gives

$$\ell_t(P_J(Y)) = r^\top B r - r^\top B Y (A + Y^\top B Y)^{-1} Y^\top B r.$$

Woodbury's identity identifies the matrix in this quadratic form as $(B^{-1} + Y A^{-1} Y^\top)^{-1} = S_{t,J}(Y)^{-1}$. It also gives

$$S_{t,J}(Y)^{-1} r = Br - BY(A + Y^\top B Y)^{-1} Y^\top B r.$$

Since A and B are diagonal, forming $Y^\top B Y$ costs $O(dk^2)$ operations, and solving the $k \times k$ system costs $O(k^3)$. □

E.2 A CONVEX SURROGATE ON EACH CUBE

The residual $r_{t,J}(Y)$ is affine in Y, but the matrix $S_{t,J}(Y)^{-1}$ changes with Y. For a fixed matrix Z, define

$$q^Z_{t,J}(Y) := r_{t,J}(Y)^\top S_{t,J}(Z)^{-1} r_{t,J}(Y).$$

This is a convex quadratic in Y: its matrix is fixed, positive definite, and applied to an affine residual. Our goal is to choose cubes small enough that replacing $S_{t,J}(Y)$ by its value at the cube center changes the loss by only a factor close to one.

Lemma E.2 (Frozen-metric sandwich)**.** *Fix a chart J and coordinate matrices Y, Z. Set*

$$\rho = \|Y - Z\|_{\mathrm{op}}, \qquad \theta = \kappa\big(2\|Z\|_{\mathrm{op}}\rho + \rho^2\big).$$

If $\theta < 1$, then

$$\frac{q^Z_{t,J}(Y)}{1+\theta} \le \ell_t(P_J(Y)) \le \frac{q^Z_{t,J}(Y)}{1-\theta} \qquad \textit{for every round } t.$$

Proof. Write $E = Y - Z$. Expanding $S_{t,J}(Y)$ around Z gives

$$S_{t,J}(Y) - S_{t,J}(Z) = E W^{-1}_{t,J} Z^\top + Z W^{-1}_{t,J} E^\top + E W^{-1}_{t,J} E^\top.$$

Since $\|W^{-1}_{t,J}\|_{\mathrm{op}} \le 1/\mu$,

$$\|S_{t,J}(Y) - S_{t,J}(Z)\|_{\mathrm{op}} \le \frac{2\|Z\|_{\mathrm{op}}\rho + \rho^2}{\mu}.$$

Also $S_{t,J}(Z) \succeq W^{-1}_{t,J^c} \succeq W^{-1} I_{d-k}$. The last two bounds show, for every vector v, that

$$\big|v^\top (S_{t,J}(Y) - S_{t,J}(Z)) v\big| \le \theta\, v^\top S_{t,J}(Z) v.$$

Consequently,

$$(1-\theta) S_{t,J}(Z) \preceq S_{t,J}(Y) \preceq (1+\theta) S_{t,J}(Z).$$

These matrices are positive definite when $\theta < 1$. Inverting the inequalities, evaluating them at $r_{t,J}(Y)$, and using the chart formula gives the claim. □

We now fix the cube resolution used by the algorithm. Let

$$D = \left\lceil \log_2 \frac{12\kappa k(d-k)}{\varepsilon} \right\rceil, \qquad s = 2^{1-D}.$$

Let $\mathcal{C}$ contain every pair (J, c) consisting of a chart J and a closed dyadic cube c of side s in its coordinate box. Closed cubes may share boundary points; this does not affect the algorithm because each cube maintains a separate learner. Write $\bar{Y}_c$ for the center of c and $q_{t,J,c} := q^{\bar{Y}_c}_{t,J}$.

Corollary E.3 (Cube resolution). *For every $(J, c) \in \mathcal{C}$, every $Y \in c$, and every round t,*

$$\frac{q_{t,J,c}(Y)}{1+\varepsilon/4} \le \ell_t(P_J(Y)) \le \frac{q_{t,J,c}(Y)}{1-\varepsilon/4}.$$

The number of chart–cube pairs is

$$N_\varepsilon := |\mathcal{C}| = \binom{d}{k} 2^{k(d-k)D} \le \binom{d}{k} \left(\frac{24\kappa k(d-k)}{\varepsilon}\right)^{k(d-k)}.$$

Proof. Each entry of $Y - \bar{Y}_c$ has magnitude at most $s/2 = 2^{-D}$. Thus

$$\rho = \|Y - \bar{Y}_c\|_{\mathrm{op}} \le \sqrt{k(d-k)}\, 2^{-D}, \qquad \|\bar{Y}_c\|_{\mathrm{op}} \le \sqrt{k(d-k)}.$$

For $Z = \bar{Y}_c$, the parameter in Lemma E.2 therefore satisfies

$$\theta \le \kappa k(d-k)\big(2^{1-D} + 4^{-D}\big) \le 3\kappa k(d-k) 2^{-D} \le \frac{\varepsilon}{4}.$$

Apply that lemma to obtain the loss comparison. Finally, $2^D < 24\kappa k(d-k)/\varepsilon$, and each chart contains $2^{k(d-k)D}$ cubes, giving the stated count. □

E.3 The algorithm and its two regret bounds

Each cube runs projected gradient descent on its frozen-metric surrogate. MWUA treats the current projector from each cube as an expert and updates its weights using the *true* WLRA losses. The full-information protocol allows both updates for every cube after each round.

Algorithm 5 Locally Convexified Grassmannian MWUA

Require: Horizon T, $1 \le k < d$, $0 < \mu \le W$, and $0 < \varepsilon \le 1$
Ensure: Rank-k plays $P_1, \dots, P_T$
  Set $\kappa = W/\mu$ and construct $D, s, \mathcal{C}, N_\varepsilon$ as above
  Set $\eta = \sqrt{8 \log N_\varepsilon / T}$, $G_{\mathrm{grad}} = 2W\sqrt{1 + k(d-k)}$, and $\Delta_{\mathrm{cube}} = s\sqrt{k(d-k)}$
  for each $(J, c) \in \mathcal{C}$ **do**
    Set $Y_1^{J,c} = \bar{Y}_c$ and $w(J, c) = 1$
  for $t = 1, \dots, T$ **do**
    Sample (J, c) proportionally to $w(J, c)$ and play $P_t = P_J(Y_t^{J,c})$
    Observe (x_t, W_t)
    for each $(J, c) \in \mathcal{C}$ **do**
      $\ell_t^{J,c} \leftarrow \ell_t(P_J(Y_t^{J,c}))$
      $w(J, c) \leftarrow w(J, c) \exp(-\eta \ell_t^{J,c}/W)$
      $g_t^{J,c} \leftarrow -2S_{t,J}(\bar{Y}_c)^{-1} r_{t,J}(Y_t^{J,c}) x_{t,J}^\top$
      $Y_{t+1}^{J,c} \leftarrow \mathrm{proj}_c\Big(Y_t^{J,c} - \frac{\Delta_{\mathrm{cube}}}{G_{\mathrm{grad}}\sqrt{t}}\, g_t^{J,c}\Big)$

Here proj_c clips each matrix entry to its interval in the closed cube c. In particular, every iterate stays inside its cube and every play remains a rank-k projector.

We first bound one cube's regret for its surrogate. The small cube diameter makes this bound proportional to ε.

Lemma E.4 (Projected gradient descent in one cube). *Fix $(J, c) \in \mathcal{C}$ and abbreviate $Y_t = Y_t^{J,c}$ and $q_t = q_{t,J,c}$. For every $Y^\circ \in c$,*

$$\sum_{t=1}^T q_t(Y_t) - \sum_{t=1}^T q_t(Y^\circ) \le \frac{3}{2} G_{\mathrm{grad}} \Delta_{\mathrm{cube}} \sqrt{T} \le \frac{3\varepsilon}{4} W\sqrt{T}.$$

Proof. The cube is convex and has Frobenius diameter Δ_{cube}. Each q_t is convex, with gradient

$$\nabla q_t(Y) = -2S_{t,J}(\bar{Y}_c)^{-1} r_{t,J}(Y) x_{t,J}^\top.$$

The spectral bounds give $\|S_{t,J}(\bar{Y}_c)^{-1}\|_{\mathrm{op}} \le W$. For $Y \in c$, the coordinate and data bounds give

$$\|r_{t,J}(Y)\|_2 \le \sqrt{1 + \|Y\|_{\mathrm{op}}^2}\, \|x_t\|_2 \le \sqrt{1 + k(d-k)}.$$

Therefore $\|\nabla q_t(Y)\|_F \le G_{\mathrm{grad}}$. Projected gradient descent with step size $\gamma_t = \Delta_{\mathrm{cube}}/(G_{\mathrm{grad}}\sqrt{t})$ has regret at most

$$\frac{\Delta_{\mathrm{cube}}^2}{2\gamma_T} + \frac{G_{\mathrm{grad}}^2}{2} \sum_{t=1}^{T} \gamma_t \le \frac{3}{2} G_{\mathrm{grad}} \Delta_{\mathrm{cube}} \sqrt{T}.$$

By the choice of D, $s \le \varepsilon/(6\kappa k(d-k))$. As $\kappa \ge 1$ and $k(d-k) \ge 1$,

$$G_{\mathrm{grad}} \Delta_{\mathrm{cube}} = 2Ws\sqrt{k(d-k)(1+k(d-k))} \le \frac{\sqrt{2}}{3}\varepsilon W \le \frac{\varepsilon W}{2}.$$

This gives the second inequality. □

Next, MWUA compares its sampled plays with the current learner in any one cube. This comparison uses true losses; it does not replace them by surrogates.

Lemma E.5 (MWUA over cubes)**.** *Let $\ell_t^{J,c} = \ell_t(P_J(Y_t^{J,c})) \in [0, W]$. For an oblivious input sequence,*

$$\mathbb{E} \sum_{t=1}^{T} \ell_t(P_t) \le \min_{(J,c)\in\mathcal{C}} \sum_{t=1}^{T} \ell_t^{J,c} + W\sqrt{\frac{T \log N_\varepsilon}{2}}.$$

Proof. Every cube's iterate depends only on earlier data, so the current loss of each cube is determined before MWUA samples a cube. Apply the standard MWUA bound to the losses $\ell_t^{J,c}/W \in [0,1]$ with uniform initial weights. The learning rate $\eta = \sqrt{8 \log N_\varepsilon / T}$ gives normalized regret at most $\log N_\varepsilon/\eta + \eta T/8 = \sqrt{T \log N_\varepsilon/2}$. Conditional on the data, the sampled play has expected loss equal to MWUA's weighted average loss. Multiplying the regret bound by W proves the claim. □

E.4 Proof of Theorem 4.2

Theorem 4.2 (Horizon-independent discretization)**.** *For $0 < \varepsilon \le 1$ and diagonal W_t under the preceding spectral bounds, Algorithm 5 satisfies*

$$\mathbb{E} \sum_{t=1}^{T} \ell_t(P_t) \le (1+\varepsilon) C_{\mathrm{WLRA}} + \varepsilon W\sqrt{T} + W\sqrt{\frac{T}{2} k(d-k) \log \frac{96\kappa k(d-k)}{\varepsilon}}.$$

Its total running time in the same exact-real model is $O\big(T(48\kappa k(d-k)/\varepsilon)^{k(d-k)}(dk^2 + k^3)\big)$.

Proof. Choose an optimal projector $P^\star$ with cumulative loss C_{WLRA}. By Lemma D.1, it has bounded chart coordinates: $P^\star = P_{J^\star}(Y^\star)$ for some $Y^\star \in [-1,1]^{(d-k)\times k}$. Choose a cube $c^\star$ containing $Y^\star$. The algorithm has a learner for this pair $(J^\star, c^\star)$.

Write Y_t and q_t for that learner's iterates and surrogates, and put $\theta = \varepsilon/4$. The iterates and $Y^\star$ all lie in $c^\star$. The cube comparison gives

$$\sum_{t=1}^{T} \ell_t^{J^\star,c^\star} \le \frac{1}{1-\theta} \sum_{t=1}^{T} q_t(Y_t).$$

The per-cube regret bound then gives

$$\sum_{t=1}^{T} q_t(Y_t) \le \sum_{t=1}^{T} q_t(Y^\star) + 3\theta W\sqrt{T}.$$

Finally, the other side of the cube comparison yields $q_t(Y^\star) \le (1+\theta)\ell_t(P^\star)$. Combining these three inequalities,

$$\sum_{t=1}^{T} \ell_t^{J^\star,c^\star} \le \frac{1+\theta}{1-\theta} C_{\mathrm{WLRA}} + \frac{3\theta}{1-\theta} W\sqrt{T}.$$

Because $\theta = \varepsilon/4 \le 1/4$, the two coefficients satisfy $(1+\theta)/(1-\theta) \le 1+\varepsilon$ and $3\theta/(1-\theta) \le \varepsilon$. Thus the chosen cube's true loss is at most $(1+\varepsilon)C_{\mathrm{WLRA}} + \varepsilon W\sqrt{T}$. Lemma E.5 adds $W\sqrt{T \log N_\varepsilon/2}$ for choosing that cube.

For the stated logarithmic bound, use $\binom{d}{k} \le 2^d$ and $d \le 2k(d-k)$ for $1 \le k < d$. Corollary E.3 then gives

$$\log N_\varepsilon \le d \log 2 + k(d-k) \log \frac{24\kappa k(d-k)}{\varepsilon} \le k(d-k) \log \frac{96\kappa k(d-k)}{\varepsilon}.$$

Substitution into the MWUA term proves the regret guarantee.

Each round updates all N_ε cubes. For one cube, Lemma E.1 computes the true loss and $S_{t,J}(\bar{Y}_c)^{-1} r_{t,J}(Y_t^{J,c})$ in $O(dk^2 + k^3)$ operations. The latter vector uses the metric at the cube center and the residual at the current iterate. The gradient, clipping, weight update, and amortized sampling fit within the same per-cube bound. Combining this cost with the cube count in Corollary E.3 proves the running time bound. The algorithm stores one $(d-k) \times k$ iterate and one MWUA weight per cube. □

F Fantope Relaxation

This appendix proves the algebraic and optimization claims used in Section 4.2.We assume the following: $\|x_t\|_2 \le 1$ and $0 \preceq W_t \preceq W I_d$. The anchor V is fixed before the online rounds. Diagonal W_t is required for the runtime stated in Theorem 4.3; the loss identities and regret proof hold for positive semidefinite W_t. In particular, the only new object in this appendix is the reduced Fantope variable $P \in \mathcal{F}_k(m)$, which is lifted to the ambient space through a fixed orthonormal anchor $V \in \mathbb{R}^{d\times m}$ by

$$\nu(P) = VPV^\top.$$

The regret guarantee below is against the best fixed fractional comparator in the reduced Fantope, not against the exact Grassmannian comparator from Section 4.1.

F.1 Lifting and Exact Loss Reduction

We first record two basic facts used by the algorithm: the lift preserves Fantope feasibility, and the ambient weighted loss can be written exactly using reduced quantities.

Lemma F.1. *If $P \in \mathcal{F}_k(m)$, then $\nu(P) = VPV^\top \in \mathcal{F}_k(d)$.*

Proof. Symmetry follows from

$$(VPV^\top)^\top = VP^\top V^\top = VPV^\top.$$

For any $y \in \mathbb{R}^d$,

$$y^\top VPV^\top y = (V^\top y)^\top P (V^\top y) \ge 0,$$

so $VPV^\top \succeq 0$. Since $P \preceq I_m$,

$$y^\top VPV^\top y \le \|V^\top y\|_2^2 \le \|y\|_2^2,$$

and thus $VPV^\top \preceq I_d$. Finally,

$$\mathrm{Tr}(VPV^\top) = \mathrm{Tr}(PV^\top V) = \mathrm{Tr}(P) = k.$$

Thus $VPV^\top \in \mathcal{F}_k(d)$. □

For the loss reduction, define

$$\widetilde{\ell}_t(P) := \|W_t^{1/2}(I_d - \nu(P))x_t\|_2^2 = \|W_t^{1/2}(I_d - VPV^\top)x_t\|_2^2,$$

and introduce

$$\alpha_t = x_t^\top W_t x_t, \qquad \bar{x}_t = V^\top x_t, \qquad \bar{z}_t = V^\top W_t x_t, \qquad \widetilde{W}_t = V^\top W_t V, \qquad \bar{X}_t = \bar{x}_t \bar{x}_t^\top.$$

Lemma F.2. *For every $P \in \mathcal{F}_k(m)$,*

$$\widetilde{\ell}_t(P) = \alpha_t - \operatorname{Tr}\left(P(\bar{z}_t \bar{x}_t^\top + \bar{x}_t \bar{z}_t^\top)\right) + \operatorname{Tr}\left(P\widetilde{W}_t P \bar{X}_t\right).$$

Proof. Let $\bar{P} = VPV^\top$. Expanding the ambient loss gives

$$\ell_t(\bar{P}) = x_t^\top W_t x_t - 2x_t^\top W_t \bar{P} x_t + x_t^\top \bar{P} W_t \bar{P} x_t.$$

The first term is α_t. For the linear term,

$$x_t^\top W_t \bar{P} x_t = x_t^\top W_t V P V^\top x_t = \bar{z}_t^\top P \bar{x}_t.$$

Since $P = P^\top$,

$$2\bar{z}_t^\top P \bar{x}_t = \operatorname{Tr}\left(P(\bar{z}_t \bar{x}_t^\top + \bar{x}_t \bar{z}_t^\top)\right).$$

For the quadratic term,

$$x_t^\top \bar{P} W_t \bar{P} x_t = \bar{x}_t^\top P (V^\top W_t V) P \bar{x}_t = \operatorname{Tr}(P\widetilde{W}_t P \bar{X}_t).$$

Combining the three identities proves the claim. □

Lemma F.3. *The reduced loss $\widetilde{\ell}_t(P)$ is convex over $\mathcal{F}_k(m)$, and*

$$\nabla \widetilde{\ell}_t(P) = -(\bar{z}_t \bar{x}_t^\top + \bar{x}_t \bar{z}_t^\top) + \widetilde{W}_t P \bar{X}_t + \bar{X}_t P \widetilde{W}_t.$$

Proof. The constant and affine terms in Lemma F.2 are convex. It remains to consider

$$H_t(P) = \operatorname{Tr}(P\widetilde{W}_t P \bar{X}_t).$$

Since $\bar{X}_t = \bar{x}_t \bar{x}_t^\top$,

$$H_t(P) = \bar{x}_t^\top P \widetilde{W}_t P \bar{x}_t = \|\widetilde{W}_t^{1/2} P \bar{x}_t\|_2^2.$$

Because $\widetilde{W}_t = V^\top W_t V \succeq 0$, this is the squared norm of a linear function of P, and is convex.

For the gradient, set

$$C_t = \bar{z}_t \bar{x}_t^\top + \bar{x}_t \bar{z}_t^\top.$$

The affine term contributes $-C_t$. For the quadratic term, write $A = \widetilde{W}_t$ and $B = \bar{X}_t$. Then

$$d \operatorname{Tr}(PAPB) = \operatorname{Tr}(dPAPB) + \operatorname{Tr}(PAdPB) = \operatorname{Tr}(dP(APB + BPA)).$$

Thus

$$\nabla H_t(P) = \widetilde{W}_t P \bar{X}_t + \bar{X}_t P \widetilde{W}_t,$$

which gives the stated expression. □

The Euclidean projection onto $\mathcal{F}_k(m)$ is spectral. If $M = M^\top$ has eigendecomposition

$$M = Q \operatorname{diag}(\lambda) Q^\top,$$

then

$$\Pi_{\mathcal{F}_k(m)}(M) = Q \operatorname{diag}(\widehat{\lambda}) Q^\top,$$

where $\widehat{\lambda}$ is the projection of λ onto the capped simplex

$$\Delta_k^{[0,1]} = \left\{ \mu \in [0,1]^m : \sum_{i=1}^m \mu_i = k \right\}.$$

Equivalently, there exists a threshold τ such that

$$\widehat{\lambda}_i = \min\{1, \max\{0, \lambda_i - \tau\}\}, \qquad \sum_{i=1}^m \widehat{\lambda}_i = k.$$

Thus projection reduces to an eigendecomposition and a capped-simplex projection.

Lemma F.4. *The reduced Fantope satisfies*

$$\sup_{P,Q\in\mathcal{F}_k(m)} \|P-Q\|_F \le \sqrt{2k}.$$

Moreover, for every $P \in \mathcal{F}_k(m)$,

$$\|\nabla\widetilde{\ell}_t(P)\|_F \le 4W.$$

Proof. For $P, Q \in \mathcal{F}_k(m)$,

$$\|P-Q\|_F^2 = \mathrm{Tr}(P^2) + \mathrm{Tr}(Q^2) - 2\mathrm{Tr}(PQ).$$

Since $0 \preceq P, Q \preceq I_m$ and both have trace k,

$$\mathrm{Tr}(P^2) \le k, \qquad \mathrm{Tr}(Q^2) \le k.$$

Also $\mathrm{Tr}(PQ) \ge 0$, so $\|P-Q\|_F^2 \le 2k$.

For the gradient bound, the affine term satisfies

$$\|\bar{z}_t\bar{x}_t^\top + \bar{x}_t\bar{z}_t^\top\|_F \le 2\|\bar{z}_t\|_2\|\bar{x}_t\|_2 \le 2W.$$

For the quadratic terms,

$$\|\widetilde{W}_t P \bar{X}_t\|_F \le \|\widetilde{W}_t\|_{\mathrm{op}}\|P\|_{\mathrm{op}}\|\bar{X}_t\|_F \le W,$$

and the same bound holds for $\|\bar{X}_t P\widetilde{W}_t\|_F$. Summing the three contributions gives $\|\nabla\widetilde{\ell}_t(P)\|_F \le 4W$. □

Theorem 4.3 (Polynomial-time fractional WLRA regret)**.** *Let V be fixed before round 1, let $\|x_t\|_2 \le 1$, and let W_t be diagonal with $0 \preceq W_t \preceq WI_d$, where $W > 0$. Running Algorithm 2 with $\eta = \Theta\left(\frac{\sqrt{k}}{W\sqrt{T}}\right)$ guarantees that for every fixed fractional comparator $P^\star \in \mathcal{F}_k(m)$, $\sum_{t=1}^T \widetilde{\ell}_t(P_t) - \sum_{t=1}^T \widetilde{\ell}_t(P^\star) \le O(W\sqrt{kT})$. Furthermore, the per-round runtime is $O(m^2d + m^3)$.*

Proof. Online gradient descent over a convex set of diameter D with gradient bound G gives, for every fixed $P^\star \in \mathcal{F}_k(m)$,

$$\sum_{t=1}^T \widetilde{\ell}_t(P_t) - \sum_{t=1}^T \widetilde{\ell}_t(P^\star) \le \frac{D^2}{2\eta} + \frac{\eta G^2 T}{2}.$$

By Lemma F.4, take

$$D = \sqrt{2k}, \qquad G = 4W.$$

With

$$\eta = \frac{\sqrt{2k}}{4W\sqrt{T}},$$

we obtain

$$\sum_{t=1}^T \widetilde{\ell}_t(P_t) - \sum_{t=1}^T \widetilde{\ell}_t(P^\star) \le 4W\sqrt{2kT}.$$

For runtime, the reduced statistics $\bar{x}_t = V^\top x_t$ and $\bar{z}_t = V^\top W_t x_t$ take $O(md)$ time, while

$$\widetilde{W}_t = V^\top W_t V$$

takes $O(m^2 d)$ time since W_t is diagonal. The gradient is computed in $O(m^2)$ time in the reduced space. The Fantope projection requires an $m \times m$ eigendecomposition, costing $O(m^3)$, plus capped-simplex projection. Thus the per-round runtime is

$$O(m^2 d + m^3).$$

□

G LOWER BOUND

We distinguish the exact-comparator and $(1+\varepsilon)$-comparator guarantees. The same offline hardness result constrains both, but its accuracy parameter is important: it gives hardness at an *inverse-polynomial* approximation scale, rather than hardness for an arbitrary fixed constant ε.

For an $m \times d$ data matrix A with rows $a_i^\top$ and a strictly positive entrywise weight matrix $H = (h_{ij})$, define

$$D_i = \mathrm{diag}(h_{i1}, \dots, h_{id}), \qquad \Phi(P) = \sum_{i=1}^{m} \min_{b \in \mathbb{R}^k} \|D_i^{1/2}(a_i - X_P b)\|_2^2.$$

The offline WLRA objective $\min_{\mathrm{rank}(Z)\le k} \sum_{i,j} h_{ij}(A_{ij} - Z_{ij})^2$ equals $\min_{P \in \mathcal{P}_k(d)} \Phi(P)$: for a fixed subspace, each row's fitting coefficients can be optimized independently, and every rank-at-most-k matrix has a row space contained in some k-dimensional subspace. Write $\mathrm{OPT} = \min_P \Phi(P)$.

Theorem G.1 (Positive-weight WLRA hardness; Gillis & Glineur (2011))**.** *For rank $k = 1$, binary $A \in \{0,1\}^{m\times d}$, and strictly positive weights $0 < h_{ij} \le 1$, it is NP-hard to produce a solution whose objective is within*

$$\gamma_{m,d} := 2^{-11}(md)^{-6}$$

of OPT. *The hard construction has weights bounded below by an inverse polynomial in md.*

The cited theorem states an *additive* objective-accuracy gap. It also yields hardness at a suitable $(1+\varepsilon)$ scale: because the zero matrix has cost at most md on these instances, $\mathrm{OPT} \le md$. Set $\varepsilon_{m,d} = \gamma_{m,d}/(2md)$. Any rank-one solution of cost at most $(1+\varepsilon_{m,d})\,\mathrm{OPT}$ would have cost at most $\mathrm{OPT} + \gamma_{m,d}/2$, contradicting Theorem G.1. In particular, the positive-weight problem has no fully polynomial-time approximation scheme unless P = NP. This argument does not establish NP-hardness for a fixed constant $\varepsilon > 0$.

We next convert a sublinear online regret guarantee into an offline approximation algorithm. Polynomial running time in the reduction below means polynomial *bit* complexity for the online updates and for evaluating the returned solutions; the exact-real arithmetic convention used to state the upper bounds does not on its own imply that complexity claim.

Theorem G.2 (Online-to-offline reduction)**.** *For an offline instance with m rows a_i and positive diagonal weight matrices D_i, write $\Phi(P) = \sum_{i=1}^m \min_b \|D_i^{1/2}(a_i - X_P b)\|_2^2$ and $\mathrm{OPT} = \min_P \Phi(P)$. Let $B = \max\{1, \max_i \|a_i\|_2\}$. Suppose that for every oblivious stream and a parameter $\sigma \ge 0$, an online algorithm satisfies*

$$\mathbb{E}\sum_{t=1}^{T} \ell_t(P_t) \le (1+\sigma)\min_P \sum_{t=1}^{T} \ell_t(P) + R_\sigma(T).$$

Then a randomized offline algorithm using T online rounds outputs a rank-k projector $\widehat{P}$ with

$$\mathbb{E}\,\Phi(\widehat{P}) \le (1+\sigma)\,\mathrm{OPT} + \frac{mB^2}{T} R_\sigma(T).$$

Suppose the online guarantee is available for every $0 < \sigma \le 1$ and

$$R_\sigma(T) \le c_\sigma T^{1-\alpha}(\log(eT))^b$$

for fixed constants $\alpha > 0$ and $b \ge 0$ independent of σ and T. Assume c_σ is independent of T and is bounded by a polynomial in the offline input size, B^2, the weight condition number, and $1/\sigma$. Assume also that the per-round bit complexity, including the representation and evaluation of the output projector, is polynomial in these parameters and T. Then, for any $\varepsilon, \delta > 0$, there is a randomized algorithm, polynomial in the same input parameters and $1/\varepsilon, 1/\delta$, that returns a solution of cost at most $(1+\varepsilon)\,\mathrm{OPT} + \delta$ with probability at least $2/3$. The same conclusion with $\varepsilon = 0$ follows if an analogous $\sigma = 0$ guarantee holds with a horizon-independent coefficient.

Proof. Let $B = \max\{1, \max_i \|a_i\|_2\}$. Draw indices $I_1, \dots, I_T$ independently and uniformly from $[m]$. At time t feed the online algorithm the normalized example $(x_t, W_t) = (a_{I_t}/B, D_{I_t})$. The unnormalized objective of a projector P is $\Phi(P)$, so its normalized expected one-round loss is

$$\mathbb{E}_{I_t}\big[\ell_t(P)\big] = \frac{\Phi(P)}{mB^2}.$$

The projector P_t is chosen before I_t is drawn. Conditional on the past and the algorithm's randomness, this identity therefore also holds with $P = P_t$. Let $P^\star$ minimize Φ. For each realized stream the online regret guarantee, with its comparator minimum bounded above by the loss of the *fixed* $P^\star$, gives

$$\mathbb{E}_{\mathcal{A}} \sum_{t=1}^{T} \ell_t(P_t) \le (1+\sigma) \sum_{t=1}^{T} \ell_t(P^\star) + R_\sigma(T).$$

Taking expectation over the sampled indices and applying conditional independence yields

$$\frac{1}{mB^2} \sum_{t=1}^{T} \mathbb{E}\, \Phi(P_t) \le \frac{(1+\sigma)T}{mB^2}\, \mathrm{OPT} + R_\sigma(T).$$

Evaluate every P_t on the entire offline instance and return $\widehat{P} \in \arg\min_t \Phi(P_t)$. Since its cost is no greater than the average of the T costs, this proves

$$\mathbb{E}\, \Phi(\widehat{P}) \le (1+\sigma)\, \mathrm{OPT} + \frac{mB^2}{T} R_\sigma(T).$$

For the high-probability statement, run the online algorithm with $\sigma = \min\{1, \varepsilon/3\}$. Choose T so that $mB^2 R_\sigma(T)/T \le \delta/3$. Because c_σ is independent of T and polynomially bounded in the stated parameters,

$$\frac{R_\sigma(T)}{T} \le c_\sigma T^{-\alpha} (\log(eT))^b.$$

For fixed $\alpha > 0$ and $b \ge 0$, such a T is polynomial in the stated parameters and $1/\varepsilon, 1/\delta$. The nonnegative random variable $\Phi(\widehat{P}) - \mathrm{OPT}$ has expectation at most $(\varepsilon\, \mathrm{OPT} + \delta)/3$. Markov's inequality gives

$$\Pr\Big[\Phi(\widehat{P}) > (1+\varepsilon)\, \mathrm{OPT} + \delta\Big] \le \frac{1}{3}.$$

For an exact-comparator online guarantee, use $\sigma = 0$ and the same argument with $mB^2 R_0(T)/T \le \delta/3$. Full evaluation of the T outputs and the assumed bit-complexity bound keep the total running time polynomial. □

Corollary G.3 (Barrier for both online guarantees)**.** *Unless* NP $\subseteq$ BPP*, there is no online family with polynomial per-round bit complexity and a sublinear regret bound as in Theorem G.2 whose coefficient is polynomial in the dimensions, the weight condition number, and* $1/\varepsilon$ *uniformly for every* $0 < \varepsilon \le 1$*. The same barrier applies to a polynomial-time sublinear-regret algorithm against the exact comparator under the analogous coefficient bound.*

Proof. Apply Theorem G.2 to the rank-one instances of Theorem G.1. On these instances $B^2 \le d$, $\mathrm{OPT} \le md$, and the condition number of the positive weights is polynomial in md. Choose offline parameters $\varepsilon = \gamma_{m,d}/(4md)$ and $\delta = \gamma_{m,d}/4$. Both inverse parameters are polynomial in md, while the resulting offline cost is at most

$$(1+\varepsilon)\, \mathrm{OPT} + \delta \le \mathrm{OPT} + \gamma_{m,d}/2 < \mathrm{OPT} + \gamma_{m,d}$$

with probability at least $2/3$. A randomized polynomial-time algorithm achieving this on the hard instances would imply NP $\subseteq$ BPP. For the exact comparator, set $\delta = \gamma_{m,d}/2$ instead. □

The exact Grassmannian construction and its locally convexified $(1+\varepsilon)$ variant both have exponential dependence on $k(d-k)$, so the corollary does not contradict either upper bound. Its conclusion is about a uniform approximation family with polynomial dependence on $1/\varepsilon$; it makes no claim that a fixed constant factor $1+\varepsilon$ is NP-hard for general positive-weight WLRA.

H Other Experiments and Details

Compute resources. Experiments were conducted on a single machine with an Intel Core i7-1165G7 CPU (4 cores, 2.70 GHz base clock, 16 GB DDR4 RAM). No GPU or cloud resources were used.

H.1 Experimental Details

Preprocessing and stream orders. For MNIST, the feature map is fitted to all $70{,}000$ images, including the first 5000 selected for evaluation. For MovieLens, the selection of movies and users and the Truncated SVD fit use the same 2000 selected users that are subsequently streamed. Both transformations are completed before stream-order permutation. The real-data experiments therefore measure online updates in fixed, transductively constructed feature spaces. The supplied runner defaults to ten reproducible permutations (seeds $0, \ldots, 9$) of each fixed dataset; these are repeated orders of the same selected observations, not ten independent datasets.

Scope and evaluation metric. For a rank-k orthonormal basis U_t, the unweighted residual is $\|x_t - U_t U_t^\top x_t\|_2^2$. The hierarchy, FTRL, and the offline rank-k reference use this form. The Fantope implementation instead evaluates $\|x_t - V P_t V^\top x_t\|_2^2$, where P_t may be fractional. The fixed random control stores k normalized but nonorthogonal vectors as columns of B and evaluates $\|x_t - BB^\top x_t\|_2^2$; its $BB^\top$ is generally not a rank-k orthogonal projector. Plots cumulate the respective per-round residual reported by each implementation. The oracle rank-k SVD solution is a lower bound for the cumulative loss of fixed rank-k projectors on this transformed stream; it is not a general lower bound for the fractional Fantope decision class or the fixed random control as implemented.

Algorithms and initialization. The adaptive hierarchy starts with the span of the first k coordinate vectors, updates its leaf centers from observed rows, and splits leaves in response to the data; it is a practical unweighted variant, not either proved Grassmannian algorithm. It uses $\eta = 0.5$, $n_{\min} = 10$, $n_{\max} = 100$, and $\varepsilon_{\mathrm{hrd}} = 0.1$. FTRL also starts with the first k coordinate vectors and updates from prior observations; its regularization parameter is $1/(t+1)$ in the implementation's update index. The fixed random control uses seed 42 to create its vectors before the first round, without access to any dataset rows, and never changes them. The offline SVD reference alone receives the entire evaluated stream before the first round.

Fantope OGD builds its anchor V by SVD of an initial stream prefix and then keeps V fixed. With anchor dimension $m = \min(d, 3k)$, the implemented prefix length is $\min\{50, \max\{10, m\}\}$: it is 30, 45, and 50 observations at $k = 10, 15, 20$, respectively, for the $d = 50$ real-data runs. Those observations are part of the reported T rounds. During this phase the implementation reports $\|x_t\|_2^2$ and includes it in cumulative loss; it makes no Fantope prediction until the prefix is processed. Every online baseline is exposed to the same observations in the same order, but uses them according to its own update rule; the fixed random control ignores them. The theorem with a fixed anchor before round 1 can be applied to the subsequent suffix with this anchor held fixed, subject to its other assumptions, but does not analyze the prefix initialization or the reported whole-stream cumulative loss. After the prefix, the implemented step size is $\sqrt{2k}/(4\sqrt{T})$.

H.2 MovieLens Dataset

Dataset and preprocessing. We use the user–movie ratings in MovieLens 20M (Harper & Konstan, 2015), retain the 1000 most frequently rated movies and then the 2000 users with the most ratings among them, and subtract 3 from observed ratings. Missing ratings are entered as zeros. Truncated SVD is fitted on this entire selected user–movie matrix to produce $d = 50$ features, after which each user's row is normalized to unit length. All evaluated users thus contribute to selecting and constructing the feature space before the streaming rounds.

Experimental setup. We stream the $T = 2000$ selected users with $k \in \{10, 15, 20\}$, using the same algorithms and hyperparameters as for MNIST. The supplied runner specifies ten seeded permutations of these users. For curves computed as pointwise averages of these runs, shaded bands denote one population standard deviation across permutations at each round.

Results. As shown in Figure 2, the hierarchical method outperforms the fixed baseline by a wide margin across all three ranks, reducing cumulative loss by 77.65% at $k = 10$, 82.76% at $k = 15$, and 86.43% at $k = 20$. Among the adaptive baselines, the hierarchical method performs competitively with FTRL, achieving lower cumulative loss at $k = 15$ and $k = 20$. In all cases, it approaches the offline optimum. These results suggest that adaptive hierarchical refinement can also identify low-dimensional structure in collaborative filtering data, where the latent preference subspace is not known in advance and must be learned online.

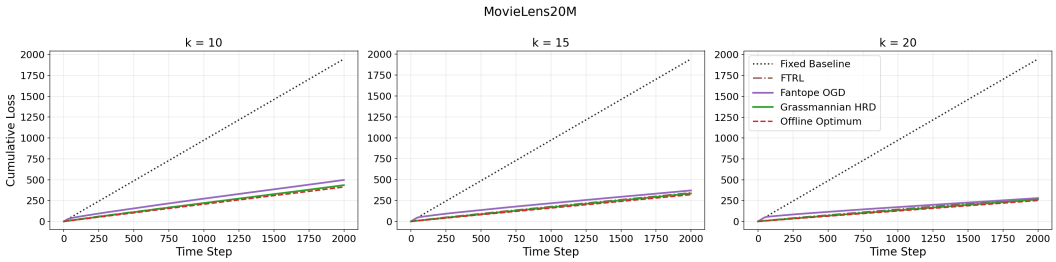

Fig. 2: Cumulative projection loss on MovieLens 20M over $T = 2{,}000$ streamed users for $k \in \{10, 15, 20\}$.

H.3 Synthetic Streams with Unequal Coordinate Reliability

The real-data runs use unweighted observations. We separately test how coordinate-dependent weights affect the Fantope implementation on synthetic streams with a known generating subspace. The implementation updates a fractional Fantope matrix using the weighted projection surrogate; these diagnostics measure the rank-k eigenspace extracted from that matrix before each update. They do not measure standard weighted reconstruction loss or the Fantope theorem's cumulative loss for its fractional decision.

Experimental setup. We generate $T = 2000$ observations in $d = 20$ with a rank-k generating subspace, $k = 3$, and use ten independently generated streams (seeds $0, \ldots, 9$). We compare weighted Fantope OGD, the same update with $W_t = I_d$, and streaming-SVD online PCA. Both Fantope variants use $m = d$ and build an anchor from their first 20 observations under the implemented rule. For the weighted and unweighted variants, let U_t consist of the top k eigenvectors of $VP_tV^\top$ before the current update. We use this extracted basis, and the streaming-SVD basis, to compute the same diagnostics as before.

$$d_{\mathrm{G}}(U_t, U^\star) = \sqrt{k - \|U_t^\top U^\star\|_F^2} = \frac{\|U_tU_t^\top - U^\star(U^\star)^\top\|_F}{\sqrt{2}}$$

$$L_t^{\mathrm{clean}} = \|s_t - U_tU_t^\top s_t\|_2^2, \quad L_t^{\mathrm{weight}} = \|W_t^{1/2}(x_t - U_tU_t^\top x_t)\|_2^2.$$

Here s_t is the clean signal in the normalized coordinate system and $U^\star$ is the generating basis. The last metric is a weighted projection residual of the extracted basis, not the fractional surrogate used to update P_t, and not the standard WLRA fitting loss for nonuniform weights. During the Fantope anchor prefix, U_t is undefined: its chordal error is omitted, while its two residual diagnostics use the zero reconstruction (giving $\|s_t\|_2^2$ and $\|W_t^{1/2}x_t\|_2^2$). Streaming SVD makes a prediction from its current basis on every round. All three methods observe the same stream, but their initialization procedures differ.

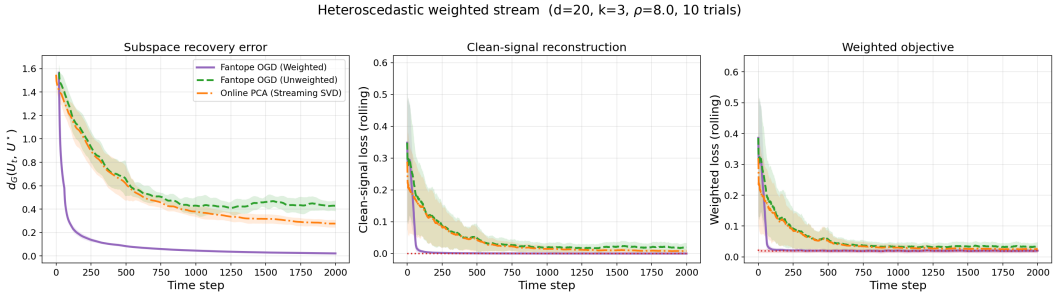

Fig. 3: Heteroscedastic stream with $d = 20$, $k = 3$, and $\rho = 8$, averaged over ten trials. The panels show chordal error relative to the generating subspace, rolling clean-signal reconstruction loss, and rolling weighted projection loss. Shaded regions show variation across trials.

Heteroscedastic noise. The generating signal lies in the reliable coordinate block; another block has larger noise. We assign inverse-variance weights, normalized so the largest is one, and set

$\rho = \sigma_{\mathrm{hi}}/\sigma_{\mathrm{lo}}$. For the trajectory plot at $\rho = 8$, $\sigma_{\mathrm{lo}} = 0.10$. In Figure 3, the weighted Fantope variant has lower extracted subspace error and clean-signal residual than the unweighted variant and online PCA. The reported final subspace errors are approximately 0.02, 0.43, and 0.28, respectively, for this setup.

We also vary ρ using $\sigma_{\mathrm{lo}} = 0.05$; hence this sweep does not hold every noise setting from the trajectory experiment fixed. In Figure 4, the weighted and unweighted Fantope variants are similar at $\rho = 1$, whereas the weighted variant has a smaller extracted subspace error as noise becomes more unequal in this tested range. Each plotted value first averages the relevant diagnostic over the final 20% of one stream and then averages that value over ten streams.

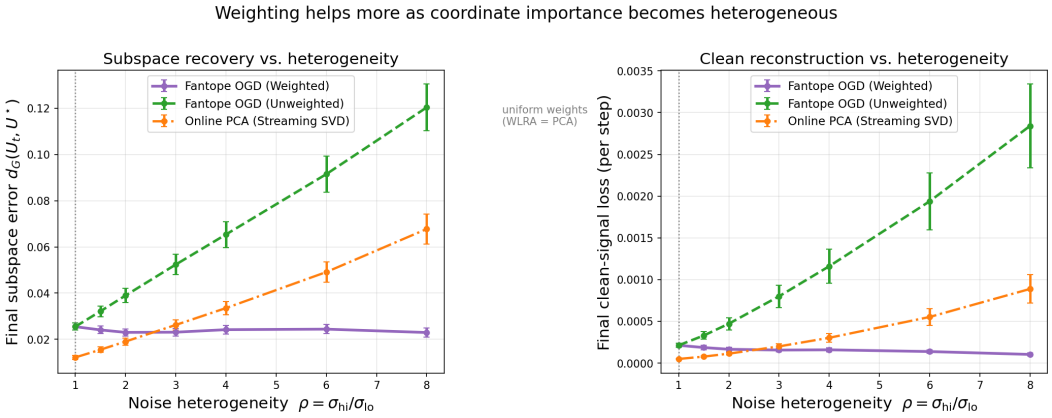

Fig. 4: Final subspace error and clean-signal reconstruction loss as the noise standard-deviation ratio $\rho = \sigma_{\mathrm{hi}}/\sigma_{\mathrm{lo}}$ varies. Points show averages over ten independently generated streams, and error bars show variation across trials. At $\rho = 1$, the coordinate weights are uniform.

Missing observations. We observe reliable-block coordinates independently with probability 0.95 and unreliable-block coordinates with probability 0.2. Missing entries are zero-filled and assigned diagonal weight 10^{-3}, while observed entries receive weight 1. Weighted Fantope OGD uses these weights; the unweighted variant and streaming SVD use the zero-filled observations without coordinate weights. In Figure 5, the weighted variant has lower extracted-basis chordal error and clean-signal residual in this setup. Its reported final values are approximately 0.53 and 0.23, versus 0.86 and 0.50 for online PCA.

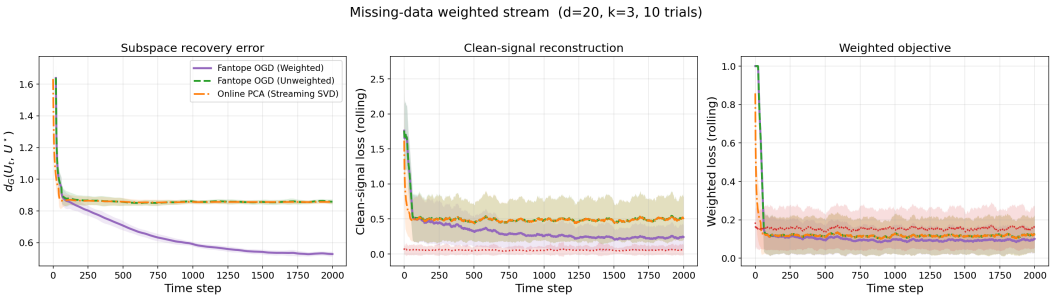

Fig. 5: Missing-observation stream with $d = 20$ and $k = 3$, averaged over ten trials. Reliable and unreliable coordinates are observed with probabilities 0.95 and 0.2, respectively. The panels show chordal subspace error, rolling clean-signal reconstruction loss, and rolling weighted projection loss.

These experiments illustrate the empirical role of the weighted Fantope surrogate when coordinates have unequal reliability. The missing-data experiment uses zero entries in W_t and therefore does not satisfy the strictly positive weight assumption used by the Grassmannian WLRA theorems.

H.4 Limitations of the Experiments

The hierarchy tested on MNIST and MovieLens uses adaptive splits and is not an implementation of either proved Grassmannian algorithm. The real-data feature spaces use the evaluated observations during preprocessing, and the fixed random control is not an orthogonal rank-k projector as implemented. The Fantope anchor is fitted from a prefix that is included in evaluation; the fixed-anchor theorem does not account for that prefix. Real-data runs reuse one selected set of MNIST images or MovieLens users under ten stream orders, whereas the synthetic plots summarize ten independently generated streams. Thus, repeated stream orders do not measure variation across new real-data samples.

The synthetic diagnostics round each fractional Fantope decision to a rank-k eigenspace for measurement. The missing-data weights are strictly positive (10^{-3} on missing entries), although these tests still evaluate the weighted projection surrogate rather than standard WLRA fitting loss. The experiments cover only the stated ranks, data selections, noise models, and hyperparameters. They illustrate the practical updates under these conditions, without directly validating the regret guarantees.