Learning and Testing Variable Partitions

Andrej Bogdanov* Baoxiang Wang†

Abstract

Let F be a multivariate function from a product set Σ^n to an Abelian group G. A k-partition of F with cost δ is a partition of the set of variables $\boldsymbol{V}$ into k non-empty subsets $(\boldsymbol{X}_1, \dots, \boldsymbol{X}_k)$ such that $F(\boldsymbol{V})$ is δ-close to $F_1(\boldsymbol{X}_1) + \cdots + F_k(\boldsymbol{X}_k)$ for some $F_1, \dots, F_k$ with respect to a given error metric. We study algorithms for agnostically learning k partitions and testing k-partitionability over various groups and error metrics given query access to F. In particular we show that

1. Given a function that has a k-partition of cost δ, a partition of cost $O(kn^2)(\delta + \varepsilon)$ can be learned in time $\tilde{O}(n^2 \operatorname{poly} 1/\varepsilon)$ for any $\varepsilon > 0$. In contrast, even for $k = 2$ and $n = 3$ learning a partition of cost $\delta + \varepsilon$ is NP-hard.
2. When F is real-valued and the error metric is the 2-norm, a 2-partition of cost $\sqrt{\delta^2 + \varepsilon}$ can be learned in time $\tilde{O}(n^5/\varepsilon^2)$.
3. When F is $\mathbb{Z}_q$-valued and the error metric is Hamming weight, k-partitionability is testable with one-sided error and $O(kn^3/\varepsilon)$ non-adaptive queries. We also show that even two-sided testers require $\Omega(n)$ queries.

This work was motivated by reinforcement learning control where the set of control variables can be partitioned. The partitioning reduces a task into multiple lower-dimensional ones that are relatively easier to learn. As a result, our second algorithm empirically increases the scores attained over previous heuristic partitioning methods.

1 Introduction

Divide-and-conquer methods rely on the ability to identify independent sub-instances of a given instance, such as connected components of graphs and hypergraphs. When these are not available one looks for partitions into loosely related parts like small or sparse cuts. These classic problems and their variants remain at the forefront of algorithmic research [KSL15, KT15, CL15, Man17, CXY18, RSW18].

We study the related problem of function decomposition: Given a multivariate function $F(\boldsymbol{V})$ over n variables $\boldsymbol{V} = \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_n\}$, we seek to partition the variables into k groups $\boldsymbol{X}_1, \dots, \boldsymbol{X}_k$ so that F decomposes into a sum $F_1(\boldsymbol{X}_1) + \cdots + F_k(\boldsymbol{X}_k)$. In case an exact decomposition of this type is unavailable, we seek an approximate one under a suitable error metric. This algebraic partitioning question can be sensibly asked for any Abelian group. While some of our results are quite general,

*`andrejb@cse.cuhk.edu.hk`. Department of Computer Science and Engineering and Institute for Theoretical Computer Science and Communications, The Chinese University of Hong Kong.

†`bxwang@cse.cuhk.edu.hk`. Department of Computer Science and Engineering, The Chinese University of Hong Kong.

two particular cases of interest are addition over $\mathbb{Z}_2$ with respect to the Hamming metric and addition over reals with respect to the 2-norm.

As a multivariate function is an exponentially large object, it is sensible to model the input F to the partitioning problem as an oracle and allow query access to it. This departs from the common setup in (hyper)graph partitioning problems, where an explicit representation of the input is assumed to be available. While variable partitioning of real-valued functions under the 2-norm turns out to be closely related to hypergraph partitioning, the difference in input access models renders certain techniques developed for the latter (e.g., random contractions) inapplicable directly to our setting.

Our work is motivated by learning control variables in high-dimensional reinforcement learning control [SB18, MBM+16, SMSM00]. If the *advantage function* of the control variables can be partitioned into multiple lower-dimensional subsets, then these subset of variables can be learned independently with a relatively easier Monte-Carlo sampling. This advantage function involves the estimates of a dynamic system, which is complex enough to not have an explicit representation available. The function is thus treated as an oracle as is in our access model. Sometimes it is natural to assume that the function should be almost decomposable; for example, if we seek to control two robots jointly performing a task, the variables controlling the respective robots are almost independent.[1] In general the dependencies are not known in advance but need to be learned from observed behavior. Some heuristic methods have been applied to control variable partitioning [WRD+18, LW18] but not rigorously analyzed.

Our contributions Our main results are algorithmic: We show that variable partitions can be learned agnostically. The precise definitions are given in Section 2.

Theorem 1. *There is an algorithm that given parameters n, k, ε, γ and oracle access to a bounded function F outputs a k-partition $\mathcal{P}$ such that $\delta(\mathcal{P}) \le O(kn^2)(\delta_k(F) + \varepsilon)$ with probability at least $1-\gamma$. The algorithm makes $O(K^p n^2 \log(n/\gamma)/\varepsilon^{2p})$ queries to F and runs in time linear in the number of queries. This holds for the p-norm over $\mathbb{R}$ measure assuming $\|F\|_{\mathbb{R},2p} = O(1)$ and for Hamming weight over $\mathbb{Z}_q$ with $p = 1$, for an absolute constant K.*

This algorithm is closely related to the heuristic ones used in the aforementioned empirical studies and works over general Abelian groups and a wide range of error metrics. However, it only guarantees optimality up to an $O(kn^2)$ approximation factor.[2] While we do not know if an approximation factor of this magnitude is inevitable, in Proposition 17 we show that obtaining a solution with additive error is NP-hard. The proofs are given in Section 4.

In contrast, our second algorithm obtains an additive error for bipartitions of real-valued functions under the 2-norm:

Theorem 2. *There is an algorithm that given inputs n, ε and oracle access to a function $F\colon \Sigma^n \to \mathbb{R}$ with $\|F\|_{\mathbb{R},4} \le 1$, runs in time $O(n^5 \log(n/\gamma)/\varepsilon^2)$ and outputs a bipartition $(\boldsymbol{X}, \overline{\boldsymbol{X}})$ such that $\delta_{\mathbb{R},2}(\boldsymbol{X}, \overline{\boldsymbol{X}})^2 \le \delta_{\mathbb{R},2}(F)^2 + \varepsilon$ with probability $1-\gamma$.*

More generally, it is possible to output a $\sqrt{2-2/k}$-approximate k-partition in time $O(n^k \operatorname{poly} nk/\varepsilon)$ (Corollary 21). For unbounded k finding a good approximation is ETH hard (Corollary 19).

Theorem 2 and Corollary 19 are based on an equivalence between variable partitioning under the 2-norm and hypergraph partitioning given in Proposition 18. The results are described and proved in Section 5.

[1] The robots may be collaborating so the decomposition might not be perfect.

[2] Recall that the input oracle has size exponential in n.

As a consequence of Theorem 1, the property of being close to a k-partition is testable with $\tilde{O}(k^{2p}n^{4p+2}/\varepsilon^{2p})$ queries. The query complexity of the tester can be somewhat improved:

Theorem 3. *k-partitionability is testable with one-sided error and $O(kn^3/\varepsilon)$ non-adaptive queries with respect to Hamming weight over $\mathbb{Z}_q$, and with $O(k^{2p}n^3/\varepsilon^{2p})$ non-adaptive queries with respect to the p-norm over $\mathbb{R}$ assuming $\|F\|_{2p} \leq 1$.*

In Section 6 we prove Theorem 3 and show that $\Omega(n)$ queries are necessary even for two-sided error testers.

Ideas and techniques Our Theorem 1 is inspired by algebraic property testing techniques. The starting point is the dual characterization of partitionability into sets $(\boldsymbol{X}, \overline{\boldsymbol{X}})$ by the constraints $D_F(\boldsymbol{X}, \boldsymbol{Y}) = 0$, where $D_F = F(X, Y) - F(X', Y) - F(X, Y') + F(X', Y')$, for all assignments X, X' to $\boldsymbol{X}$ and Y, Y' to $\boldsymbol{Y}$. David et al. [DDG+17] apply this relation to random inputs towards testing whether a $\mathbb{Z}_2$-valued function F tensors decomposes into a direct sum. The acceptance probability of this test approximates the best decomposition to within a factor of 4 (Proposition 4).

Our partitioning algorithm estimates the *dependence score* $\|D_F(\boldsymbol{x}, \boldsymbol{y})\|$ on every pair of variables $\boldsymbol{x}, \boldsymbol{y}$ (keeping the rest fixed) to decide whether they should be partitioned or not. Here, $\|T\|$ is the probability that the test T fails for discrete groups like $\mathbb{Z}_2$. In general it can represent any error metric satisfying the axioms in Section 2. The proof of Theorem 1 amounts to showing that a collection of single variable partitions $(\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{P}$ for which the local scores $\|D_F(\boldsymbol{x}, \boldsymbol{y})\|$ are small can be glued together into a single k-partition $\mathcal{P}$ with a small global score.

When F is real-valued and error is measured under the 2-norm, variable partitioning has a natural geometric interpretation. Functions that depend on different coordinates are orthogonal modulo their constant term, so the optimal decomposition with respect to a fixed partition $(\boldsymbol{X}_1, \ldots, \boldsymbol{X}_k)$ is given by the projection of F onto the respective subspaces of functions. This yields an equality between the distance and the dependence score (4) for bipartitions and a generalization to k-partitions (Proposition 8). Variable partitioning for functions is then equivalent to hypergraph parititioning of their orthogonal decompositions (Proposition 18), with the cost of cut $(\boldsymbol{X}, \overline{\boldsymbol{X}})$ given by $\frac{1}{4}\|D_F(\boldsymbol{x}, \boldsymbol{y})\|^2$.

This connection suggests the application of hypergraph partitioning algorithms that can be implemented with access to an *approximate* cut oracle[3], leading to Theorem 2. On the negative side it reveals that approximately optimal partitions into a large number of components are hard to find (Corollary 19).

Application to reinforcement learning control We plug our partitioning algorithm back to reinforcement learning control. In this setting, the oracle is real-valued and as we adapt the 2-norm we use the submodularity cut algorithm described in Theorem 2.

We compare empirically with three previous approaches: The baseline that does not involve partitioning [MBM+16, Wil92]; the baseline that trivially partitions n variables into n subsets [WRD+18, Kos18]; the work that partitions the variables heuristically [LW18]. The way [LW18] partitions the variables is to calculate the discrete estimate of the Hessian of the oracle. Then they

[3]Several state-of-the art algorithms for cuts in graphs and hypergraphs rely on random contractions [Kar00, KS96, CXY18]. In particular, Rubinstein et al. [RSW18] showed that $\tilde{O}(n)$ queries to an *exact* cut oracle and similar running time are sufficient to find the minimum cut. We do not know if comparable efficiency can be obtained with an approximate oracle.

remove from Hessian the elements with lowest absolute values, until it forms at least k connected components if the Hessian matrix is treated as the adjacency matrix.

The scores we attained on the tasks in the physics simulator are improve over these approaches, which is demonstrated in Section 7.

2 Formulation of Variable Partitioning

Let $F(\boldsymbol{V})$ be a function from some product set to an abelian group G. A direct sum decomposition of F is a partition $(\boldsymbol{X}_1, \dots, \boldsymbol{X}_k)$ of the set of variables $\boldsymbol{V}$ such that

$$F(\boldsymbol{V}) = F_1(\boldsymbol{X}_1) + \cdots + F_k(\boldsymbol{X}_k) \tag{1}$$

for some functions $F_1, \dots, F_k$. We are interested in efficient (sublinear-time) approximate algorithms for computing the direct sum decomposition given oracle access to F.

Assume the variables $\boldsymbol{V}$ take values in some set Σ endowed with a product measure. Given a partition $\boldsymbol{X}_1, \dots, \boldsymbol{X}_k$ of the variables, the quality of the decomposition with respect to this partition is measured by

$$\delta(\boldsymbol{X}_1, \dots, \boldsymbol{X}_k) = \min_{F_1,\dots,F_k} \|F(X_1, \dots, X_k) - F_1(X_1) - \cdots - F_k(X_k)\|,$$

where $\|\cdot\|\colon G \to \mathbb{R}^+$ is a partial norm satisfying the following three axioms:

1. $\|0\| = 0$;
2. $\|F + G\| \leq \|F\| + \|G\|$;
3. $\mathbb{E}[\|F(X, \cdot)\| \mid X] \leq \|F\|$ for any set of variables X of F.

Examples of interest G include the group $\mathbb{Z}_q$ under the Hamming metric $\|F\| = \Pr[F(V) = 0]$ and the group $\mathbb{R}$ under the p-norm $\|F\|_p = \mathbb{E}\big[|F(V)|^p\big]^{1/p}$ for any $p \geq 1$.

We seek an approximation of the best-possible partition, which minimizes the objective

$$\delta_2(F) = \min \delta(\boldsymbol{X}, \overline{\boldsymbol{X}}),$$

where the minimum is taken over all bipartitions $(\boldsymbol{X}, \overline{\boldsymbol{X}})$ of $\boldsymbol{V}$ and $\overline{\boldsymbol{X}}$ denotes the complement of $\boldsymbol{X}$. More generally, we consider partitions into (at least) k nonempty sets, giving rise to the objective value $\delta_k(F) = \min \delta(\boldsymbol{X}_1, \dots, \boldsymbol{X}_k)$.

Our algorithms are based on the following dependence estimator inspired by the rank-1 test of [DDG+17]. Let $\boldsymbol{X}$ and $\boldsymbol{Y}$ be two disjoint sets of variables. The dependence estimator $D_F(\boldsymbol{X}, \boldsymbol{Y})$ is the random variable

$$D_F = F(X, Y, Z) + F(X', Y', Z) - F(X', Y, Z) - F(X, Y', Z)$$

where X, X' are independent samples of the $\boldsymbol{X}$ variable, Y, Y' are independent samples of the $\boldsymbol{Y}$ variable, and Z is a random sample of the remaining variables. If F decomposes into a direct sum that partitions the $\boldsymbol{X}$ and $\boldsymbol{Y}$ variables then D_F equals zero. Conversely, $\|D_F\|$ measures the quality of the approximation.

In the analysis it will be convenient to use the notation $F \approx_\delta G$ for $\|F - G\|_p \leq \delta$. The following two facts are immediate consequences of axioms 2 and 3:

Triangle inequality: If $F \approx_\delta G$ and $G \approx_{\delta'} H$ then $F \approx_{\delta+\delta'} H$.

Fixing: If $F(X,Z) \approx_\delta G(X,Z)$ then $F(\underline{X},Z) \approx_\delta G(\underline{X},Z)$ for some fixed value $\underline{X}$.

Some of our results are specific to $\mathbb{R}$ under the 2-norm in which case we use the notation $\delta_{\mathbb{R},2}$ and $\|\cdot\|_{\mathbb{R},2}$.

3 Estimating the quality of a partition

In this section we show that $\|D_F(\boldsymbol{X},\boldsymbol{Y})\|$ is an approximate estimator for the quality $\delta(\boldsymbol{X},\boldsymbol{Y})$ of a decomposition, namely

$$\delta(\boldsymbol{X},\boldsymbol{Y}) \le \|D_F(\boldsymbol{X},\boldsymbol{Y})\| \le 4\cdot\delta(\boldsymbol{X},\boldsymbol{Y}). \tag{2}$$

The proof is given in Claims 5 and 6 below. As $\|D_F(\boldsymbol{X},\boldsymbol{Y})\|$ can be estimated efficiently from oracle access to F (Claim 7), we obtain an algorithm for estimating the quality of a partition to within a factor of 4 in general, and exactly for the 2-norm over $\mathbb{R}$.

Proposition 4. *There is an algorithm that given a bipartition $\boldsymbol{X},\boldsymbol{Y}$ of the variables and parameters $\varepsilon,\gamma > 0$, outputs a value $\hat{\delta}$ such that*

$$\delta(\boldsymbol{X},\boldsymbol{Y}) \le \hat{\delta} \le 4\cdot\delta(\boldsymbol{X},\boldsymbol{Y}) + \varepsilon,$$

with probability at least $1-\gamma$ from $K^p\log(1/\gamma)/\epsilon^{2p}$ queries to F in time linear in the number of queries. This holds for the p-norm over $\mathbb{R}$ measure assuming $\|F\|_{\mathbb{R},2p} = O(1)$ and for Hamming weight over $\mathbb{Z}_q$ with $p=1$, for an absolute constant K.

The value of $\delta(\boldsymbol{X},\boldsymbol{Y})$ is known to be NP-hard to calculate exactly over $\mathbb{Z}_2$ under the Hamming metric given explicit access to the truth-table of f [RV07]. Therefore some approximation factor is unavoidable for algorithms running in time polynomial in n and $1/\varepsilon$ unless BPP is in NP. On the positive side Karpinski and Schudy [KS09] give a fully polynomial-time randomized approximation scheme for this special case.

The analysis in fact applies to any pair of disjoint subsets $\boldsymbol{X}$, $\boldsymbol{Y}$ that do not necessarily partition all the variables. In this more general setting distance is measured by the formula

$$\delta(\boldsymbol{X},\boldsymbol{Y}) = \min_{A,B} \|F(X,Y,Z) - A(X,Z) - B(Y,Z)\|. \tag{3}$$

Claim 5 (Completeness of D_F)**.** *For all disjoint $\boldsymbol{X}$, $\boldsymbol{Y}$, $\|D_F(\boldsymbol{X},\boldsymbol{Y})\| \le 4\cdot\delta(\boldsymbol{X},\boldsymbol{Y})$.*

Proof. By definition of $\delta(\boldsymbol{X},\boldsymbol{Y})$ there exists a decomposition of the form

$$F(X,Y,Z) = A(X,Z) + B(Y,Z) + D(X,Y,Z),$$

where $\|D(X,Y,Z)\| = \delta(\boldsymbol{X},\boldsymbol{Y})$. In the expansion of D_F all the A and B terms cancel out, leaving

$$\begin{aligned}\|D_F(\boldsymbol{X},\boldsymbol{Y})\| &= \|D(X,Y,Z) + D(X',Y',Z) - D(X,Y',Z) - D(X',Y,Z)\| \\ &\le \|D(X,Y,Z)\| + \|D(X',Y',Z)\| + \|D(X,Y',Z)\| + \|D(X',Y,Z)\| \\ &= 4\delta(\boldsymbol{X},\boldsymbol{Y}).\end{aligned}$$

□

Soundness for Boolean functions under the uniform measure was proved by David et al. [DDG+17]. We reproduce their proof under a more general setting.

Claim 6 (Soundness of D_F)**.** *For all disjoint $\boldsymbol{X}$, $\boldsymbol{Y}$, $\delta(\boldsymbol{X}, \boldsymbol{Y}) \le \|D_F(\boldsymbol{X}, \boldsymbol{Y})\|$.*

Proof. Let $\varepsilon = \|D_F(\boldsymbol{X}, \boldsymbol{Y})\|$. Then

$$F(X, Y, Z) \approx_\varepsilon F(X, Y', Z) - F(X', Y, Z) - F(X', Y', Z).$$

We can fix values $\underline{X}'$ and $\underline{Y}'$ for which

$$F(X, Y, Z) \approx_\varepsilon F(\underline{X}', \underline{Y}', Z) - F(X, \underline{Y}', Z) - F(\underline{X}', Y, Z) = A(X, Z) + B(Y, Z),$$

where $A(X, Z) = F(\underline{X}', \underline{Y}', Z) - F(X, \underline{Y}', Z)$ and $B(Y, Z) = F(\underline{X}', Y, Z)$. □

Proposition 4 now follows from inequality (2) and the following claim, which states that $\|D_F\|$ can be estimated by sampling in the cases of interest. See Appendix A for the proof.

Claim 7. *Assuming $\|F\|_{\mathbb{R},2p} \le 1$, the value $\|F\|_{\mathbb{R},p}$ can be estimated within ε from $K^p \log(1/\gamma)/\epsilon^{2p}$ (random) queries to F in linear time with probability $1 - \gamma$ for some absolute constant K.*

3.1 Exact partitioning under the 2-norm

Since computing the optimal partition is in general NP-complete, we do not expect to replace the inequalities in (2) with an equality. However, in the special case of real-valued functions with ℓ^2-norm, the estimate becomes exact:

$$\|D_F(\boldsymbol{X}, \boldsymbol{Y})\|_{\mathbb{R},2} = 2 \cdot \delta_{\mathbb{R},2}(\boldsymbol{X}, \boldsymbol{Y}). \tag{4}$$

This equality is a consequence of the following characterization of $\delta_{\mathbb{R},2}$, which applies more generally to k-partitions:

Proposition 8. *Assuming $\mathbb{E}[F] = 0$, the k-partition $F_i(X_i) = \mathbb{E}[F|X_i]$ achieves the minimum for $\delta_{\mathbb{R},2}(\boldsymbol{X}_1, \dots, \boldsymbol{X}_k)$.*

In particular it follows that $\delta_{\mathbb{R},2}$ takes the value

$$\delta_{\mathbb{R},2}(\boldsymbol{X}_1, \dots, \boldsymbol{X}_k) = \mathbb{E}[(\overline{F} - \mathbb{E}[\overline{F}|X_1] - \cdots - \mathbb{E}[\overline{F}|X_k])^2], \tag{5}$$

where $\overline{F} = F - \mathbb{E}[F]$. To derive identity (4) it remains to verify that when $k = 2$, the right-hand side of (5) is a quarter of $\|D_F\|^2$:

Fact 9. $\|D_F(\boldsymbol{X}, \boldsymbol{Y})\|_{\mathbb{R},2}^2 = 4 \cdot \mathbb{E}[(\overline{F} - \mathbb{E}[\overline{F}|X] - \mathbb{E}[\overline{F}|Y])^2]$.

Armed with this fact we prove the proposition.

Proof of Proposition 8. First assume $F(\boldsymbol{X}, \boldsymbol{Y})$ is bivariate. Let $A(\boldsymbol{X})$ be any function. The inequality $\mathbb{E}[(\mathbb{E}[F|X] - A(X))^2] \ge 0$ can be rewritten as

$$\mathbb{E}[(F - \mathbb{E}[F|X])^2] \le \mathbb{E}[(F - A(X))^2], \tag{6}$$

stating that the orthogonal projection of F onto the subspace of functions that depend only on $\boldsymbol{X}$ in 2-norm is $\mathbb{E}[F|X]$.

Now let $F(\boldsymbol{X}_1, \ldots, \boldsymbol{X}_k)$ be k-variate. Assume $\mathbb{E}[F] = 0$ and $\mathbb{E}[F_i(X_i)] = 0$ for all i. Then $\mathbb{E}[F_i(X_i)|X_j] = 0$ for all $i \neq j$. Applying inequality (6) for k times in succession together with this fact, we obtain

$$\begin{aligned}
&\mathbb{E}[(F - F_1(X_1) - \cdots - F_{k-1}(X_{k-1}) - F_k(X_k))^2] \\
&\qquad \geq \mathbb{E}[(F - F_1(X_1) - \cdots - F_{k-1}(X_{k-1}) - \mathbb{E}[F - F_1(X_1) - \cdots - F_{k-1}(X_{k-1})|X_k])^2] \\
&\qquad = \mathbb{E}[(F - F_1(X_1) - \cdots - F_{k-1}(X_{k-1}) - \mathbb{E}[F|X_k])^2] \\
&\qquad \vdots \\
&\qquad \geq \mathbb{E}[(F - \mathbb{E}[F|X_1] - \cdots - \mathbb{E}[F|X_k])^2]
\end{aligned}$$

as desired. Finally, by orthogonality the optimal decomposition must satisfy $\sum \mathbb{E}[F_i(X_i)] = 0$ so the assumption $\mathbb{E}[F_i(X_i)] = 0$ can be made without loss of generality. $\square$

By orthogonality, equation (5) can also be written in the following forms:

$$\begin{aligned}
\delta_{\mathbb{R},2}(\boldsymbol{X}_1, \ldots, \boldsymbol{X}_k) &= \mathbb{E}\left[\overline{F}^2\right] - \sum\nolimits_{i=1}^k \mathbb{E}\left[\mathbb{E}[\overline{F}|X_i]^2\right] \\
&= \mathbb{E}_X\left[\overline{F}(X)^2\right] - \sum_{i=1}^k \mathbb{E}_{X,X'}\left[\overline{F}(X_{-i}, X_i)\overline{F}(X'_{-i}, X_i)\right],
\end{aligned} \tag{7}$$

where (X_{-i}, X_i) is the input whose i-th variable takes value X_i and j-th variable takes value X'_j for $j \neq i$. As all these terms can be efficiently estimated, we obtain the following algorithm for estimating the quality of a given k-partition:

Proposition 10. *There is an algorithm that given a k-partition $\boldsymbol{X}_1, \cdots, \boldsymbol{X}_k$ of the variables and parameters $\varepsilon, \gamma > 0$, outputs a value $\hat{\delta}$ such that*

$$|\hat{\delta}^2 - \delta_{\mathbb{R},2}(\boldsymbol{X}_1, \ldots, \boldsymbol{X}_k)^2| \leq \varepsilon,$$

with probability at least $1 - \gamma$ from $O(k \log(k/\gamma)/\epsilon^4)$ queries to F in time linear in the number of queries.

4 Variable partitioning over general groups

In this section we present our first partitioning algorithm, which is general enough to work on any normed group G.

The algorithm is based on the pairwise estimates of dependency over sets of single variables. The intuition behind the algorithm is that if the dependency between $\boldsymbol{x}$ and $\boldsymbol{y}$ is low, then these two variables should be assigned to different partitions. Therefore the algorithm keeps asserting such "in different partitions" for the pairs with lowest dependency estimates, until the k-partitioning can be clearly observed from the assertions.

This idea of the algorithm has been used in previous works in reinforcement learning control [WRD+18, LW18] in a heuristic way. Theorem 1 below shows that the algorithm outputs an $O(kn^2)$ approximation to the optimal partition in time polynomial in n, k, and $1/\varepsilon$.

Proposition 11. *Assuming $e(\boldsymbol{x}, \boldsymbol{y}) \leq \hat{e}(\boldsymbol{x}, \boldsymbol{y}) \leq e(\boldsymbol{x}, \boldsymbol{y}) + \varepsilon$ for all $\boldsymbol{x}$ and $\boldsymbol{y}$,*

$$\delta(\mathcal{P}) \leq (8k - 10)n^2(4\delta_k(F) + \varepsilon). \tag{8}$$

If the estimates $\hat{e}(\boldsymbol{x}, \boldsymbol{y})$ are obtained by empirical averaging, we obtain Theorem 1.

Algorithm 1 Approximate partitioning via pairwise estimates

Input: number of partitions k
Output: partition $\mathcal{P}$
For every pair of variables $\boldsymbol{x}, \boldsymbol{y} \in \boldsymbol{V}$, find estimate $\hat{e}(\boldsymbol{x}, \boldsymbol{y})$ for $e(\boldsymbol{x}, \boldsymbol{y}) = \|D_F(\{\boldsymbol{x}\}, \{\boldsymbol{y}\})\|$;
Create a weighted graph with vertices $\boldsymbol{V}$ and weights $\hat{e}(\boldsymbol{x}, \boldsymbol{y})$;
Order the edges in increasing weight;
repeat
    Remove the edge with the smallest weight;
until The graph has exactly k connected components

4.1 Proof of Proposition 11

For a partition $\mathcal{P}$ of the variables, let $\Delta(\mathcal{P}) = \sum \delta(\{\boldsymbol{x}\}, \{\boldsymbol{y}\})$, where the sum is taken over all pairs that cross the partition. We will deduce Theorem 1 from the following bound on $\delta(\mathcal{P})$.

Claim 12. *For every k-partition $\mathcal{P}$, $\delta(\mathcal{P}) \le (16k - 20)\Delta(\mathcal{P})$.*

The following fact is immediate from the definitions of δ. The proof of this claim is delayed to the end of this section.

Fact 13. *For any partition $(\boldsymbol{U}, \overline{\boldsymbol{U}})$ such that $\boldsymbol{X} \subseteq \boldsymbol{U}$ and $\boldsymbol{Y} \subseteq \overline{\boldsymbol{U}}$, $\delta(\boldsymbol{X}, \boldsymbol{Y}) \le \delta(\boldsymbol{U}, \overline{\boldsymbol{U}})$.*

Now we prove the theorem, assuming the correctness of Claim 12.

Proof of Theorem 1. By Claim 5 and Fact 13, all edges $(\boldsymbol{x}, \boldsymbol{y})$ in the optimal partition must satisfy $e(\boldsymbol{x}, \boldsymbol{y}) \le 4\delta_2(F)$. By our assumption on the quality of the approximations,

$$\hat{e}(\boldsymbol{x}, \boldsymbol{y}) \le 4\delta_2(F) + \varepsilon. \tag{9}$$

Since the algorithm removes edges in increasing order of weight, all the edges that cross the output partition $\mathcal{P}$ must also satisfy this inequality. Then

$$\begin{aligned}
\delta(\mathcal{P}) &\le (16k - 20)\Delta(\mathcal{P}) && \text{by Claim 12,} \\
&\le (16k - 20) \sum\nolimits_{\boldsymbol{x}, \boldsymbol{y} \text{ cross } \mathcal{P}} e(\boldsymbol{x}, \boldsymbol{y}) && \text{by Claim 6,} \\
&\le (16k - 20) \sum\nolimits_{\boldsymbol{x}, \boldsymbol{y} \text{ cross } \mathcal{P}} \hat{e}(\boldsymbol{x}, \boldsymbol{y}) && \\
&\le (16k - 20) \sum\nolimits_{\boldsymbol{x}, \boldsymbol{y} \text{ cross } \mathcal{P}} 4\delta_2(F) + \varepsilon && \text{by (9),} \\
&\le (8k - 10)n^2 \cdot (4\delta_2(F) + \varepsilon).
\end{aligned}$$

The last inequality holds because there are at most $\binom{n}{2} \le n^2/2$ pairs of variables crossing the partition. □

We first prove Claim 12 in the case $k = 2$ of bipartitions. This is Claim 15 below. We use $\boldsymbol{X}\boldsymbol{X}'$ to denote the union of the variable sets $\boldsymbol{X}$ and $\boldsymbol{X}'$.

Claim 14. *For disjoint sets of variables $\boldsymbol{X}, \boldsymbol{X}', \boldsymbol{Y}$, $\delta(\boldsymbol{X}\boldsymbol{X}', \boldsymbol{Y}) \le \delta(\boldsymbol{X}, \boldsymbol{Y}) + 2\delta(\boldsymbol{X}', \boldsymbol{Y})$.*

Proof. Assume that

$$F(X, X', Y) \approx_\delta A(X, X') + B(X', Y) \qquad \text{and}$$
$$F(X, X', Y) \approx_{\delta'} A'(X, X') + B'(X, Y).$$

By the triangle inequality,

$$A(X, X') + B(X', Y) \approx_{\delta+\delta'} A'(X, X') + B'(X, Y).$$

Fix $X'(Z) = \underline{X}'(Z)$. Writing $C(X) = A(X, \underline{X}') - A'(X, \underline{X}')$ and $D(Y') = B(\underline{X}', Y')$ we get that

$$B'(X, Y) \approx_{\delta+\delta'} C(X) + D(Y).$$

By the triangle inequality (with the second equation), we get that

$$F(X, X', Y) \approx_{\delta+2\delta'} A'(X, X') + C(X) + D(Y). \qquad \square$$

Claim 15. *For every bipartition $\boldsymbol{X}, \overline{\boldsymbol{X}}$ of the variables, $\delta(\boldsymbol{X}, \overline{\boldsymbol{X}}) \le 4 \cdot \Delta(\boldsymbol{X}, \overline{\boldsymbol{X}})$.*

Proof. By Claim 14,

$$\delta(\boldsymbol{X'}\{\boldsymbol{x}\}, \{\boldsymbol{y}\}) \le \delta(\boldsymbol{X'}, \{\boldsymbol{y}\}) + 2\delta(\{\boldsymbol{x}\}, \{\boldsymbol{y}\})$$

for all $\boldsymbol{X'} \subseteq \boldsymbol{X} \setminus \{x\}$ and $\boldsymbol{y}$. Applying this inequality iteratively we conclude that $\delta(\boldsymbol{X}, \{\boldsymbol{y}\}) \le 2\sum_{\boldsymbol{x}\in\boldsymbol{X}} \delta(\{\boldsymbol{x}\}, \{\boldsymbol{y}\})$. Also by Claim 14

$$\delta(\boldsymbol{X}, \boldsymbol{Y}'\{\boldsymbol{y}\}) \le \delta(\boldsymbol{X}, \boldsymbol{Y}') + 2\delta(\boldsymbol{X}, \boldsymbol{Y}'\{\boldsymbol{y}\}),$$

so $\delta(\boldsymbol{X}, \boldsymbol{Y}) \le 2\sum_{\boldsymbol{y}\in\boldsymbol{Y}} \delta(\boldsymbol{X}, \{\boldsymbol{y}\})$. Combining the two inequalities we obtain the desired conclusion. $\square$

To extend the proof to larger k and obtain Claim 12, we generalize the first inequality in this sequence to k-partitions.

Claim 16. *For every $2k$-partition $(\boldsymbol{Y}_1, \dots, \boldsymbol{Y}_k, \boldsymbol{Z}_1, \dots, \boldsymbol{Z}_k)$,*

$$\delta(\boldsymbol{Y}_1, \dots, \boldsymbol{Y}_k, \boldsymbol{Z}_1, \dots, \boldsymbol{Z}_k) \le 2\delta(\boldsymbol{Y}_1\boldsymbol{Z}_1, \dots, \boldsymbol{Y}_k\boldsymbol{Z}_k) + 3\delta(\boldsymbol{Y}_1 \dots \boldsymbol{Y}_k, \boldsymbol{Z}_1 \dots \boldsymbol{Z}_k).$$

Proof. Assume that

$$F(V) \approx_\delta F_1(Y_1, Z_1) + \cdots + F_t(Y_t, Z_t)$$
$$F(V) \approx_{\delta'} A(Y_1, \dots, Y_t) + B(Z_1, \dots, Z_t).$$

By the triangle inequality

$$A(Y_1, \dots, Y_t) + B(Z_1, \dots, Z_t) \approx_{\delta+\delta'} F_1(Y_1, Z_1) + \cdots + F_t(Y_t, Z_t).$$

Fixing $Z_1, \dots, Z_t$ to values $\underline{Z}_1, \dots, \underline{Z}_t$ we get the decomposition

$$A(Y_1, \dots, Y_t) \approx_{\delta+\delta'} F_1(Y_1, \underline{Z}_1) + \cdots + F_t(Y_t, \underline{Z}_t) - B(\underline{Z}_1, \dots, \underline{Z}_t).$$

and similarly

$$B(Z_1, \dots, Z_t) \approx_{\delta+\delta'} F_1(\underline{Y}_1, Z_1) + \cdots + F_t(\underline{Y}_t, Z_t) - A(\underline{Y}_1, \dots, \underline{Y}_t).$$

Plugging these into the second equation gives the desired decomposition. $\square$

Proof of Claim 12. We assume that k is a power of two and prove by induction that $\delta(\mathcal{P}) \leq c_k \Delta(\mathcal{P})$, where c_k is the sequence $c_{2k} = 2c_k + 12$, $c_2 = 4$. The base case $k = 2$ follows from Claim 15. Assume the claim holds for k and apply Claim 16 to $\mathcal{P}$. By inductive assumption and Claim 15,

$$\delta(\mathcal{P}) \leq 2 \cdot c_k \Delta(\boldsymbol{Y}_1\boldsymbol{Z}_1, \dots, \boldsymbol{Y}_k\boldsymbol{Z}_k) + 3 \cdot 4\Delta(\boldsymbol{Y}_1 \dots \boldsymbol{Y}_k, \boldsymbol{Z}_1 \dots \boldsymbol{Z}_k).$$

Since $\mathcal{P}$ is a refinement of both these partitions, it follows that $\delta(\mathcal{P}) \leq (2c_k + 12)\Delta(\mathcal{P}) = c_{2k}\Delta(\mathcal{P})$, concluding the induction.

The recurrence solves to $c_k = 8k - 12$, proving the claim when k is a power of two. When it is not, the same reasoning applies to the closest power of two exceeding k (by taking some of the sets in the partition to be empty), which is at most $2k - 1$, proving the desired bound. □

4.2 Hardness of exact variable bipartitioning

In contrast to Theorem 1, finding the exact bipartition is NP-hard even when there are only three variables.

Proposition 17. *For any $n \geq 3$ there is no algorithm that outputs a bipartition of cost $\delta_2(F) + \varepsilon$ over $\mathbb{Z}_2$ under Hamming distance in time polynomial in $|\Sigma|/\varepsilon$ with constant probability unless BPP is in NP.*

Proof. Assume such an algorithm *BIPARTITION* exists. We show it can be used to solve the following problem: Given explicit functions $F_1, \dots, F_{n-1} \colon \Sigma^2 \to G$, find i^* that minimizes $\delta_2(F_{i^*})$ assuming this i^* is unique, i.e. a function that has the smallest bipartition cost among the candidates.

Let $F(x_1, \dots, x_{n-1}, y) = F_1(x_1, y) + \cdots + F_{n-1}(x_{n-1}, y)$. The cost of the bipartition that splits x_i from all other variables in F is at most the cost of F_i, so F has a bipartition of cost at least $\delta_2(F_{i^*})$. We now argue that the cost of all other bipartitions is greater. In fact, any other bipartition must split y from x_i for some $i \neq i^*$. Assuming the cost of this partition is δ, we must have

$$F(X, Y) \approx_\delta A(X) + B(Y),$$

where $x_i \in X$ and $y \in Y$. After fixing all variables except for x_i and y we obtain

$$F_i(x_i, y) \approx_\delta A(x_i, \underline{X_{-i}}) + B(y, \underline{Y_{-i}}) - \sum_{j \neq i} F_j(\underline{x_j}, y).$$

This is a bipartition for F_i so its cost is strictly greater than $\delta_2(F_{i^*})$. Therefore the output of *BIPARTITION* with oracle access to F and $\varepsilon = 1/|\Sigma|^2$ has the desired property.

Roth and Viswanathan [RV07] give an efficient reduction R that maps a graph G into a function F such that if G has a larger max-cut than G' then $\delta_2(R(G)) < \delta_2(R(G'))$. By composing the two reductions we obtain an efficient algorithm for deciding which of two graphs has a larger maximum cut, which is an NP-hard problem. □

5 Partitioning real-valued functions under the 2-norm

The problem of partitioning real-valued functions under the 2-norm is closely related to the well-studied problem of hypergraph partitioning. To explain this connection we recall the Efron-Stein decomposition of real-valued functions over product sets. The Efron-Stein decomposition of a function $F\colon \Sigma^n \to \mathbb{R}$ (under some product measure) is the unique decomposition of the form

$$F(x) = \sum\nolimits_{S \subseteq [n]} \hat{F}_S \cdot F_S(x),$$

where $\hat{F}_S$ are real coefficients and F_S are functions satisfying the following properties:

1. F_S depends on the variables in S only;
2. $\mathbb{E}[F_S | x_{-i}] = 0$, where x_{-i} is a fixing of all variables except the i-th one;
3. $\mathbb{E}[F_S^2] = 1$.

In particular, properties 1 and 2 imply that $\mathbb{E}[F_S F_T] = 0$ when $S \neq T$, and so $\mathbb{E}[F^2] = \sum_S \hat{F}_S^2$ by property 3.

Proposition 18. *Given $F\colon \Sigma^n \to \mathbb{R}$, let H be the hypergraph whose vertices are the variables of F and whose hyperedges S have weight $\hat{F}_S^2$ for every subset S. The cost of the k-cut $(\boldsymbol{X}_1, \ldots, \boldsymbol{X}_k)$ in H equals $\delta_{\mathbb{R},2}(\boldsymbol{X}_1, \ldots, \boldsymbol{X}_k)^2$.*

Proof. We may assume $\mathbb{E}[F] = 0$ and use expression (7) to evaluate $\delta_{\mathbb{R},2}$. The first term equals $\mathbb{E}[F^2] = \sum_S \hat{F}_S^2$. The rest of the terms have the form $\mathbb{E}[\mathbb{E}[F|X_I]^2] = \mathbb{E}[F(X_{-I}, X_I) F(X'_{-I}, X_I)]$ for subsets I of variables. Plugging in the Efron-Stein decomposition of F we have

$$\mathbb{E}[\mathbb{E}[F|X_I]^2] = \sum_{S,T} \hat{F}_S \hat{F}_T \mathbb{E}[F_S(X_{-I}, X_I) F_T(X'_{-I}, X_I)].$$

By property 2, the terms in the summation in which $S \neq T$ evaluate to zero. Among the rest, if the set S contains any variable i outside I then

$$\mathbb{E}[F_S(X_{-I}, X_I) F_T(X'_{-I}, X_I)] = \mathbb{E}\big[F_S(X_{-I}, X_I) \mathbb{E}[F_T(X'_{-I}, X_I) | X, X'_{-i}]\big]$$

and the inside expectation evaluates to zero by property 2. Therefore the only surviving terms are those where $S = T$ and $S \subseteq I$, from where

$$\mathbb{E}[\mathbb{E}[F|X_I]^2] = \sum\nolimits_{S \subseteq I} \hat{F}_S^2.$$

By (7),

$$\delta_{\mathbb{R},2}(\boldsymbol{X}_1, \ldots, \boldsymbol{X}_k) = \sum\nolimits_S \hat{F}_S^2 - \sum\nolimits_{i=1}^k \sum\nolimits_{S \subseteq \boldsymbol{X}_i} \hat{F}_S^2 = \sum\nolimits_{S \not\subseteq \boldsymbol{X}_i \text{ for any } i} \hat{F}_S^2.$$

The last quantity is the desired value of the cost of k-cut in H. $\square$

When $\Sigma = \{-1, 1\}$ under uniform measure, the functions F_S do not depend on F and equal the Fourier characters $\chi_S(x) = \prod_{i \in S} x_i$, allowing us to embed instances of hypergraph partitioning into variable partitioning.

Corollary 19. *Assume there is an algorithm A that given oracle access to $F\colon \{-1,1\}^n \to \mathbb{R}$ under uniform measure outputs a k-variable partition of cost at most $C \cdot \delta_2(F) + \varepsilon$ in time $t(n, k, \varepsilon)$. Then given a hypergraph with n vertices and m hyperedges with a k-cut of value opt, it is possible to output a k-cut of value $C \cdot opt$ in time $mnt(n, k, 1/m)$.*

Chekuri and Li [CL15] give a reduction from hypergraph k-cut to densest-k-subgraph. Manurangsi [Man17] shows that the latter is hard to approximate to within $O(n^{1/(\log\log n)^c})$ assuming the exponential-time hypothesis, implying inapproximability of the same order for $\delta_{\mathbb{R},2}(F)$.

On the positive side, Proposition 18 can be used to obtain variable partitioning algorithms from hypergraph partitioning ones. The conversion is not direct as hypergraph partitioning assumes explicit access to the hypergraph. Klimmek and Wagner [KW96] observed that submodularity of the hypergraph cut function $f(\boldsymbol{X}) = \delta_{\mathbb{R},2}(\boldsymbol{X}, \overline{\boldsymbol{X}})^2$ allows for efficient minimization from *exact* oracle access. To derive Theorem 2 we extend the analysis to approximate oracle access.

The following proposition is an analysis of Queyranne's symmetric submodular minimization algorithm [Que98] for an approximate input oracle. We say g is ε-submodular if $g(\boldsymbol{XYZ}) - g(\boldsymbol{XZ}) - g(\boldsymbol{YZ}) + g(\boldsymbol{Z}) \le \varepsilon$ for all disjoint subsets $\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}$.

Proposition 20 (Queyranne's algorithm with an approximate oracle)**.** *There is an algorithm that given oracle access to an ε-submodular g, makes $O(n^3)$ oracle queries and outputs a nontrivial subset $\boldsymbol{X}$ such that $g(\boldsymbol{X})$ is within $n\varepsilon/2$ of the minimum of g.*

Proof of Theorem 2. By Proposition 10, $\delta_{\mathbb{R},2}(\boldsymbol{X}, \overline{\boldsymbol{X}})^2$ can be estimated to within error ε/Kn with $O(\log(n/\gamma)n^2/\varepsilon^2)$ queries to F with probability $1 - K\gamma/n^3$ for any constant K. This estimator implements an $\varepsilon/2n$-approximate oracle to $\delta_{\mathbb{R},2}(\boldsymbol{X}, \overline{\boldsymbol{X}})^2$ with probability $1-\gamma$ with respect to an algorithm that makes at most Kn^3 queries. In particular, with probability $1-\gamma$, the output of the oracle is $\varepsilon/4n$-close to the value of the submodular function $\delta^2_{\mathbb{R},2}$ at all points queried by Queyranne's algorithm and also at the minimum of $\delta^2_{\mathbb{R},2}$. Since from the algorithm's perspective it is interacting with a ε/n-submodular function g, it outputs a partition such that $g(\boldsymbol{X}, \overline{\boldsymbol{X}})$ is within $\varepsilon/2$ of the minimum of g. By the triangle inequality, $\delta_{\mathbb{R},2}(\boldsymbol{X}, \overline{\boldsymbol{X}})^2$ is within $\varepsilon/2 + 2\varepsilon/4n \le \varepsilon$ close to the minimum of $\delta^2_{\mathbb{R},2}$. □

Saran and Vazirani's approximation algorithm for multiway k-cut (with fixed terminals) can be viewed as a reduction from multiway k-cut to multiway 2-cut. The reduction works given access to approximate cut oracles and can therefore be used to compute k-partitions of variables.

Corollary 21. *There is an algorithm that given inputs k, ε and oracle access to a function F such that $\|F\|_{\mathbb{R},4} \le 1$ runs in time $O(k^2 n^{k+5}\log(n/\gamma)/\varepsilon^2)$ and outputs a k-partition $\mathcal{P}$ such that $\delta_{\mathbb{R},2}(\mathcal{P})^2 \le (2-2/k)\delta_{\mathbb{R},2}(F)^2 + \varepsilon$ with probability $1-\gamma$.*

It remains to prove Proposition 20.

Claim 22. *Let g be ε-submodular. Assume there exists $\boldsymbol{x} \in \boldsymbol{W}$ such that for all $\boldsymbol{Y} \subseteq \boldsymbol{W} \setminus \boldsymbol{x}$ and $\boldsymbol{u} \notin \boldsymbol{W}$,*

$$g(\boldsymbol{W}) + g(\boldsymbol{u}) \le g(\boldsymbol{W} \setminus \boldsymbol{Y}) + g(\boldsymbol{Yu}) + \delta.$$

If $\boldsymbol{x}'$ maximizes $g(\boldsymbol{Wu}) - g(\boldsymbol{u})$ among all $\boldsymbol{u} \notin \boldsymbol{W}$ then

$$g(\boldsymbol{Wx}') + g(\boldsymbol{u}) \le g(\boldsymbol{Wx}' \setminus \boldsymbol{Y}) + g(\boldsymbol{Yu}) + (\delta + \varepsilon).$$

Proof. If $\boldsymbol{x} \notin \boldsymbol{Y}$ then

$$\begin{aligned} g(\boldsymbol{Wx}') + g(\boldsymbol{u}) &\le \big(g(\boldsymbol{W}) - g(\boldsymbol{W} \setminus \boldsymbol{Y}) + g(\boldsymbol{Wx}' \setminus \boldsymbol{Y})\big) + g(\boldsymbol{u}) + \varepsilon && \text{by } \varepsilon\text{-submodularity} \\ &= g(\boldsymbol{Wx}' \setminus \boldsymbol{Y}) + \big(g(\boldsymbol{W}) - g(\boldsymbol{W} \setminus \boldsymbol{Y}) + f(\boldsymbol{u})\big) + \varepsilon \\ &\le g(\boldsymbol{Wx}' \setminus \boldsymbol{Y}) + g(\boldsymbol{Yu}) + (\delta + \varepsilon) && \text{by inductive hypothesis.} \end{aligned}$$

Otherwise, $\boldsymbol{x} \notin \boldsymbol{W} \setminus \boldsymbol{Y}$ and

$$
\begin{aligned}
g(\boldsymbol{W}\boldsymbol{x}') + g(\boldsymbol{u}) &\leq g(\boldsymbol{W}\boldsymbol{u}) + g(\boldsymbol{x}') && \text{by optimality of } \boldsymbol{x}' \\
&\leq \big(g(\boldsymbol{W}) - g(\boldsymbol{Y}) + g(\boldsymbol{Y}\boldsymbol{u})\big) + g(\boldsymbol{x}') + \varepsilon && \text{by } \varepsilon\text{-submodularity} \\
&= \big(g(\boldsymbol{W}) + g(\boldsymbol{x}') - g(\boldsymbol{Y})\big) + g(\boldsymbol{Y}\boldsymbol{u}) + \varepsilon \\
&\leq g(\boldsymbol{W}\boldsymbol{x}' \setminus \boldsymbol{Y}) + g(\boldsymbol{Y}\boldsymbol{u}) + (\delta + \varepsilon) && \text{by inductive hypothesis.}
\end{aligned}
$$
□

Proof of Proposition 20. Queyranne's algorithm Q^g is recursive. If $n = 2$ the unique partition is output. Otherwise, starting from an arbitrary singleton set $\boldsymbol{W}_1$, the algorithm sets $\boldsymbol{W}_{i+1} = \boldsymbol{W}_i\boldsymbol{x}_i$, where $\boldsymbol{x}_i$ maximizes $g(\boldsymbol{W}_i\boldsymbol{u}) - g(\boldsymbol{u})$ among all $\boldsymbol{u} \notin \boldsymbol{W}_i$. The algorithm then contracts the elements $\boldsymbol{x}_{n-1}$ and $\boldsymbol{x}_n$ into $\boldsymbol{x}_{n-1}\boldsymbol{x}_n$ and outputs the smaller value of $Q^g(\boldsymbol{x}_1, \dots, \boldsymbol{x}_{n-2}, \boldsymbol{x}_{n-1}\boldsymbol{x}_n)$ and $g(\boldsymbol{x}_n)$.

We prove by induction on n that the output of Q^g is $(n-1)\varepsilon/2$-close to the minimum of g. The base case $n = 2$ is clear. Now assume this is true for inputs of size $n - 1$. If the mimimum partition of g doesn't split $\boldsymbol{x}_{n-1}$ and $\boldsymbol{x}_n$ then the claim follows by inductive assumption.

Otherwise, we show that $g(\boldsymbol{x}_n)$ is within $(n-1)\varepsilon/2$-close to the minimum of g. Applying Claim 22 iteratively to the sets $\boldsymbol{W}_1, \dots, \boldsymbol{W}_{n-1}$, we conclude that

$$g(\boldsymbol{W}_{n-1}) + g(\boldsymbol{x}_n) \leq g(\boldsymbol{W}_{n-1} \setminus \boldsymbol{Y}) + g(\boldsymbol{Y}\boldsymbol{x}_n) + (n-1)\varepsilon$$

for all $\boldsymbol{Y}$ that do not contain $\boldsymbol{x}_n$ and $\boldsymbol{x}_{n-1}$. Applying symmetry this inequality can be rewritten as $g(\boldsymbol{x}_n) \leq g(\boldsymbol{x}_n Y) + (n-1)\varepsilon/2$. As $\boldsymbol{x}_{n-1}$ and $\boldsymbol{x}_n$ are split in the optimal solution it must be of type $\boldsymbol{x}_n Y$ for some Y excluding $\boldsymbol{x}_{n-1}$, so $g(\boldsymbol{x}_n)$ is $(n-1)/2\varepsilon$ close to the minimum as desired. □

6 Testing partitionability

As a consequence of Theorem 1, k-partitionability is testable with $\tilde{O}(k^{2p}n^{4p+2}/\varepsilon^{2p})$ queries. The yes instances are inputs with $\delta_k(F) = 0$; the no instances are inputs with $\delta_k(F) > \varepsilon$. The query complexity of Theorem 3 can be obtained by the improved analysis that follows.

To simplify notation we only prove the theorem for the Hamming weight over $\mathbb{Z}_q$ and describe the change necessary for p-norms over $\mathbb{R}$.

Algorithm 2 Tester for k-partitionability

```
1: Create an empty undirected graph G with vertex set V.
2: for O(kn/ε) times do
3:     for For every pair of distinct variables x, y ∈ V do
4:         Choose random x, y, x', y', Z
5:         If F(x, y, Z) + F(x', y', Z) ≠ F(x, y', Z) + F(x', y, Z) create the edge {x, y} in G.
6:     end for
7: end for
8: Accept if the graph has at least k connected components.
```

We will need the following fact:

Fact 23. *If $p_1, \dots, p_n$ are probabilities such that $\sum p_i \geq \varepsilon$ then $1 - \prod(1 - p_i) \geq \varepsilon - O(\varepsilon^2)$.*

Proof. Using the inequality $1 - p \leq e^{-p}$ and the second-order estimate $e^{-x} = 1 - x + O(x^2)$, we have

$$\prod(1 - p_i) \leq \prod e^{-p_i} = e^{-\sum p_i} \geq e^{-\varepsilon} = \varepsilon - O(\varepsilon^2).$$
□

Proof of Theorem 3. The connected components of G are always contained in the partition components of F, so if F is k partitionable the tester always accepts. We argue that with constant probability, all bipartition of F satisfying $\delta(\boldsymbol{X}, \overline{\boldsymbol{X}}) \geq \varepsilon$ cross an edge in G.

If F is ε-far from k-partitionable, by Claim 12 $\Delta(\mathcal{P}) = \Omega(\varepsilon/k)$ for all k-partitions $\mathcal{P}$. As every k-cut can be coarsened into a 2-cut of at least half the weight, every k-parititon can be coarsened into a bipartition such that $\Delta(\boldsymbol{X}, \overline{\boldsymbol{X}}) = \Omega(\varepsilon/k)$. We now argue that with constant probability, all such heavy bipartitions $(\boldsymbol{X}, \overline{\boldsymbol{X}})$ are crossed by an edge in G, so no k-partition is likely to survive in G.

Assume $\Delta(\boldsymbol{X}, \overline{\boldsymbol{X}}) = \sum_{\boldsymbol{x} \in \boldsymbol{X}, \boldsymbol{y} \in \boldsymbol{X}} \|D_F(\boldsymbol{x}, \boldsymbol{y})\| = \Omega(\varepsilon/k)$. As $\|D_F(\boldsymbol{x}, \boldsymbol{y})\|$ is the acceptance probability of the test in line 5, by Fact 23 in any given iteration of the outer loop 3 at least one of these edges will appear in G with probability $\Omega(\varepsilon/k)$. (For p-norms over $\mathbb{R}$, $\|D_F(\boldsymbol{x}, \boldsymbol{y})\|$ is an expectation that takes $O((\varepsilon/k)^{2p})$ queries to estimate.) After $O(kn/\varepsilon)$ iterations the probability that the cut survives is less than 2^{-n}. By a union bound the probability that any heavy cut survives is at most half. □

It is not difficult to see that n queries are required for one-sided error testers as the relevant constraints span an n-dimensional space. We show that a linear dependence on n is necessary for two-sided error testers as well. Proposition 24 shows a general $\Omega(n)$ lower bound for functions over finite domains (with uniform measure) valued over finite groups under the Hamming metric. Proposition 25 shows that $\Omega(n)$ *non-adaptive* queries are necessary for functions from $\mathbb{R}^n$ to $\mathbb{R}$ under the 2-norm.

Proposition 24. *Testing 2-partitionability for functions $F\colon \mathbb{Z}_q^n \to G$ for a finite group G under uniform measure and Hamming metric requires $\Omega(n)$ queries even for constant ε.*

For simplicity of notation we present the proof in the case $q = 2$ and $G = \mathbb{Z}_2$.

Proof. Let $R\colon \mathbb{Z}_2^n \to \mathbb{Z}_2$ be a random function and $P\colon \mathbb{Z}_2^{n-1} \to \mathbb{Z}_2$ be a function that depends on all but a random hidden input coordinate I. First we argue that $\delta_2(R) = \Omega(1)$ with high probability. For this it is sufficient to argue that $\|D_R(\boldsymbol{X}, \boldsymbol{Y})\| = \Omega(1)$ for every partition $(\boldsymbol{X}, \boldsymbol{Y})$. By definition $\|D_R(\boldsymbol{X}, \boldsymbol{Y})\|$ is the average value of $\Omega(2^{2n})$ indicator values for events of the type $R(x, y) + R(x', y') - R(x', y) - R(x, y') = 0$. These events have probability half each and are pairwise independent, so by Chebyshev's inequality the probability that the $\|D_R(\boldsymbol{X}, \boldsymbol{Y})\|$ is sub-constant is $\Omega(2^{-2n})$. Taking a union bound over all 2^n bipartitions it follows that $\|D_R(\boldsymbol{X}, \boldsymbol{Y})\| = \Omega(1)$ with probability at least $1 - \Omega(2^{-n})$.

To complete the proof, it is sufficient to argue that with high probability any Q queries to P reveal independent random bits. Consider the subspace of $\mathbb{Z}_2^n$ spanned by the Q queries (or the submodule of $\mathbb{Z}_q^n$ if q is not a prime). This vector space has dimension at most Q, so it can contain at most Q of the elementary basis vectors $e_1, \dots, e_n$. However, unless it contains e_I for the hidden coordinate I, no two queries differ in a single coordinate and all answers are independent random bits. Since I is uniformly random the probability that P and R can be distinguished is at most Q/n. By a union bound the distinguishing advantage of the tester is at most $\Omega(2^{-n}) + Q/n$, which is subconstant unless $Q = \Omega(n)$. □

It is unclear if the proof of Proposition 24 can be extended to functions over the reals as there is no uniform measure over the class of functions $\mathbb{R}^n \to \mathbb{R}$. Proposition 25 shows that testing still requires $\Omega(n)$ non-adaptive queries in this setting.

Proposition 25. *Testing k-partitionability non-adaptively for functions from $\mathbb{R}^n$ to $\mathbb{R}$ under the 2-norm requires $\Omega(n)$ queries under any measure with zero mean and unit variance and bounded third and fourth moments.*

We need the following claim about distinguishing linear functions of normal random variables.

Claim 26. *Let $Z_1, \ldots, Z_n$ be independent standard normal random variables, $F(x) = \sum_{i=1}^n Z_i x_i$, and $F'(x) = \sum_{i \in S} Z_i x_i$ where $S \subseteq [n]$ is a random subset of size s. For any q queries $x^1, \ldots, x^q \in \mathbb{R}^n$, $(F(x^1), \ldots, F(x^q))$ and $(F'(x^1), \ldots, F'(x^q))$ are $O(qs/(n-s+1))$-statistically close.*

Proof. It is sufficient to prove the claim for $|S| = n-1$ and apply the triangle inequality. By convexity it is sufficient to upper bound the expected statistical distance averaged over the choice of the index i that is omitted from S. For fixed i, since the queried functions are linear, without loss of generality we may assume that the queries $x^1, \ldots, x^q$ are orthonormal. Let X be the $q \times n$ matrix whose rows are the queries $x^1, \ldots, x^n$, and X_{-i} be the submatrix obtained by removing the i-th column. The desired statistical distance is then within a constant factor of $\|(X^T X)^{-1}(X_{-i}^T X_{-i}) - I\|_F$, where $\|\cdot\|_F$ is the Frobenius norm [BU87]. By orthonormality we obtain that $\|(X^T X)^{-1}(X_{-i}^T X_{-i}) - I\|_F = \|x^i\|_2^2$. Averaging over i, the desired statistical distance is at most $O(\mathbb{E}_i[\|x^i\|_2^2]) = O(q/n)$. □

Proof of Proposition 25. Let $F(x) = n^{-1} \sum_{j \neq k} Z_{jk} x_j x_k$, where Z_{jk} are independent standard normal random variables. Let $F'(x) = n^{-1} \sum_{j,\,k,\,i \text{ distinct}} Z_{jk} x_j x_k$ where i is chosen at random from $[n]$. By standard concentration inequalities both $\|F\|_4$ and $\|F'\|_4$ are constant with high probability. By Claim 26, the answers to any q non-adaptive queries to F and F' are $O(q/n)$-statistically close.

It remains to argue that F is $\Omega(1)$-far from 2-partitionable. For a fixed bipartition (S, T) of $[n]$, by Claim 26 the cost of F is $n^{-1} \sum_{j \in S, k \in T} Z_{jk}^2$. Therefore the average cost (over the randomness of F) is $|S||T|/n$. By Chernoff bound the cost is at least $\Omega(|S||T|/n) = \Omega(1)$ with probability $1 - \exp(-\Omega(|S||T|))$. Taking a union bound over all 2^{n-1} possible bipartitions we conclude that F is $\Omega(1)$-far from 2-partitionable with probability $1 - \exp(-\Omega(n))$. □

If Claim 26 extends to adaptive queries, so would Proposition 25.

7 Applications to reinforcement learning control

In this section we discuss the application of variable partitioning algorithms given real-valued oracle and the 2-norm measure. The set $\boldsymbol{X}$ of variables to be partitioned corresponds to the set a of control variables (the action), while the oracle F corresponds to the advantage function A. While the control task is achieved by a series of actions, the advantage function describes how much one single action in the series can affect the final objective. In general, this function is complex enough so the explicit representation is not available. Instead, it is usually estimated by Monte-Carlo sampling of the action series or by function approximation, where in either case it is sensible to treat the function as an oracle.

In a reinforcement learning control task, the objective is to control the action a so as to maximize the expected cumulative reward over time t. The advantage function $A(\cdot, a)$ describes the marginal gain of such an objective of an action $a = a_t$ at time t. This function can be estimated by Monte-Carlo sampling of the actions $a_t, a_{t+1}, \ldots$, or by function approximation. In either of the cases it is sensible to treat the function as an oracle when using it to partition the variables.

We compare empirically with three previous approaches. The first approach is a standard approach proposed by Williams [Wil92, SB18] and later improved by Mnih et al. [MBM+16] and Schulman et al. [SWD+17]. These approaches learns reinforcement learning control without considering the possible partitioning of the advantage function. The second approach is to trivially partition n variables into n subsets [WRD+18, Kos18]. This causes large partition error which induces bias in the learning update. The third baseline partitions the variables heuristically [LW18]. In their method the Hessian matrix of the advantage function is first calculated using a discrete gradient method. Then this Hessian matrix is treat as an adjacency matrix of a graph, by the heuristic that two independent variables are have a zero element in Hessian. Then elements are removed from Hessian, from those with the lowest absolute values, until the graph has at least k connected components. This algorithm shares some similar intuition with our first algorithm.

The rest of this section will introduce the preliminaries of how this partition may be used in reinforcement learning, and then demonstrate the comparison of scores attained in the experiments.

7.1 Reinforcement learning control and policy gradient

We consider a reinforcement learning task described by a discrete-time Markov decision process (MDP), denoted as the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{T}, r, \rho_0, \beta)$. That includes $\mathcal{S} \in \mathbb{R}^m$ the m dimensional state space, $\mathcal{A} \in \mathbb{R}^n$ the n dimensional action space, $\mathcal{T} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}^+$ the environment transition probability function, $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ the reward function, ρ_0 the initial state distribution and $\beta \in [0, 1)$ the unnormalized discount factor. Here n is the number of the control variables, which is consistent with the dimension of the input oracle. A (stochastic) policy is a function $\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}_+$ that outputs a distribution over $\mathcal{A}$ on any given state $s \in \mathcal{S}$. The objective of reinforcement learning is to learn a policy π such that the expected cumulative reward $J(\theta) = \mathbb{E}_{s \sim \rho_\pi, a \sim \pi}[\sum_{t=0}^{\infty} \beta^t r(s_t, a_t)]$, is maximized, where $\rho_\pi(s) = \sum_{t=1}^{\infty} \beta^{t-1} \mathbb{P}(s_t = s)$. Since π is in a functional space, the problem is commonly relaxed to find over the space of parameterized functions the policy, such as the space of neural networks. When the policy is parameterized we denote it as π_θ.

Advantage actor-critic (A2C), a standard approach in policy optimization [MBM+16, SWD+17], estimates the gradient of the policy $\nabla_\theta J(\theta)$. According to the policy gradient theorem [Wil92], this gradient can be estimated by $\nabla_\theta J(\theta) = \mathbb{E}_{\pi(a|s)}[\nabla_\theta \log \pi(a|s) A^\pi(s, a)]$, where $A^\pi(s, a)$ is the advantage function of s,a, and policy π. Here $A^\pi(s, a)$ is defined as $A^\pi(s, a) = Q^\pi(s, a) - V^\pi(s)$, where $Q^\pi(s, a) = \mathbb{E}_\pi[\sum_{t' \geq t}^{\infty} \beta^{t'-t} r(s_{t'}, a_{t'}) | s = s_t, a = a_t, \pi]$ is the action-state value function and $V^\pi(s) = \mathbb{E}_{a \sim \pi(a|s)}[Q^\pi(s, a)]$ the state-value function.

It is shown later in [WRD+18] and [LW18], that an alternative estimator

$$\nabla_\theta J(\theta) = \sum_{j=1}^{k} \mathbb{E}_{\pi(a_{(j)}|s)}[\nabla_\theta \log \pi(a_{(j)}|s)(A^\pi(s, a_{(j)})], \tag{10}$$

may induces a lower variance. The condition that this estimator holds is that the advantage function can be approximately partitioned into k parts correspondingly:

$$A^\pi(s, a) = A_1^\pi(s, a_{(1)}) + \cdots + A_k^\pi(s, a_{(k)}) + U(s, a)$$

for some state s the estimation takes place, where $U(s, a)$ the partition error is expected to be small for the estimator to be accurate.

The learning is a iterative process that takes N updates by the gradient $\nabla_\theta J(\theta)$ while the k-partition is computed every N/N_1 iterations. Every run of the partitioning algorithm outputs the disjoint subsets $a_{(1)}, \ldots, a_{(k)}$, which is then used by (10) for N_1 iterations. It is worth note that our algorithm has a complexity of $\mathcal{O}(N_1 n^5)$, which is negligible in reinforcement learning. As the Monte-Carlo estimation of $\nabla_\theta J(\theta)$ requires a complete trial of the task (for example, play a game for an entire episode), which involves the interaction of a complex system.

7.2 Experiments

We compare our first algorithm (called pairwise estimates - PE) and our second algorithm (called submodular minimization - SM) with the aforementioned existing approaches. The comparisons is summarized below.

PG estimator	Variance	Heuristics	Partitioning	Guarantees	Limits
A2C [MBM+16]	CV	-	-	-	-
Wu et al. [WRD+18]	CV & RB	yes	fully	no	$k = m$
Li and Wang [LW18]	CV & RB	yes	greedy	no	**no**
PE (our first)	CV & RB	**no**	greedy	factor-$\mathcal{O}(kn^2)$	**no**
SM (our second)	CV & RB	**no**	**optimal**	**almost opt**	**no**

Table 1: Comparisons of our algorithms with previous ones

Now we study the performance (in terms of both the correctness and the optimality) on graph cuts on weighted graphs. This will illustrate the difference between greedy-based algorithms like [LW18] and our first algorithm, and submodular minimization based algorithms like our second algorithm. Note that submodular minimization always finds the optimal partition.

#Nodes n	$n = 5$	$n = 10$	$n = 20$	$n = 40$	$n = 100$
Submodular	-	-	-	-	-
Greedy (correctness)	7753	6271	4226	2380	1101
Greedy (optimality)	1.060	1.203	1.408	1.352	1.250

Table 2: Performance of the greedy algorithm on variable partition

Then Table 3 compares the partitioning algorithms when the oracle is a quadratic function $a^T Ha$ for some random H. In this case our second algorithm SM also incurs an error per Theorem 2, but the error in practice is shown to be small enough. It has constantly the best empirical performance in both correctness and optimality.

Since we only replaced heuristic partitioning with our partitioning algorithm in reinforcement learning, it is reasonable that our more accurate partitions will improve reinforcement learning.

Finally we plug our algorithms into reinforcement learning control, replacing the partitioning steps in [LW18]. The tasks we are testing on are standard tasks in reinforcement learning by the MuJoCo physics simulator. This includes training a simplified model of ant, cheetah, or human to run as fast as possible. The score is the cumulative reward over time, where the reward is the speed less the energy cost (which is $0.001\|a\|_2^2$). The control variables a are the forces applied on the joints. We refer to [BCP+16] for the exact simulator settings.

We have conducted experiments on all eight environments from MuJoCo that has the action dimensional higher than one, shown in Figure 1 below. In the figure x-axis is the number of

#Nodes n	$n=5$	$n=10$	$n=20$	$n=40$	$n=100$
Li and Wang [LW18] (correctness)	7553	5651	2929	1251	400
PE (correctness)	7709	6108	4001	2020	918
SM (correctness)	**9896**	**9630**	**9243**	**8193**	**6802**
Li and Wang [LW18] (optimality)	1.150	1.281	1.508	1.501	1.290
Wu et al. [WRD+18] (optimality)	9.049	13.54	20.96	34.42	72.55
PE (optimality)	1.075	1.277	1.452	1.400	1.281
SM (optimality)	**1.020**	**1.028**	**1.101**	**1.110**	**1.025**

Table 3: Comparisons of the algorithms on variable partition

Monte-Carlo sample updates, which can be regarded as the time elapsed on the training, while y-axis is the score attained by the model. Our second algorithm (SM) has achieved the highest score among most of these tasks, which agrees with our theoretical finding.

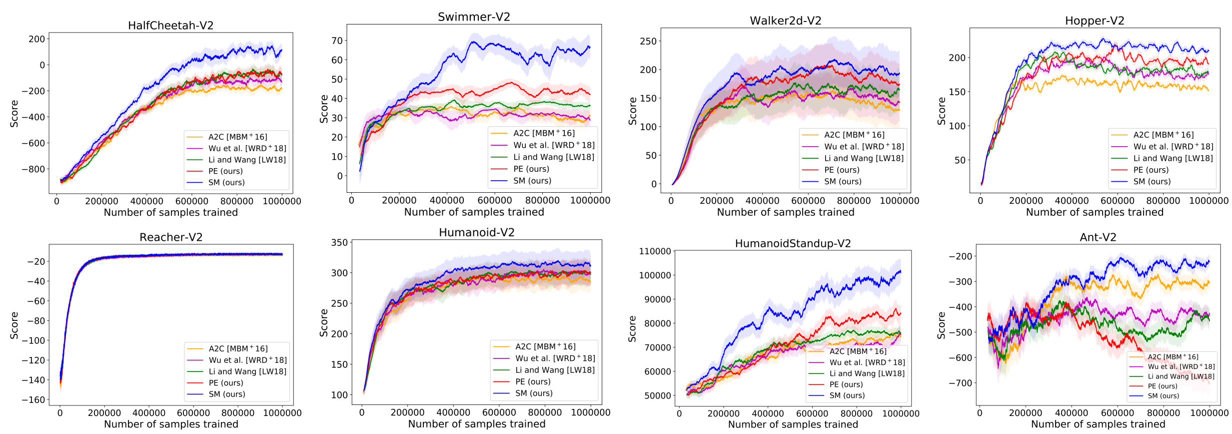

Figure 1: Empirical comparisons on MuJoCo high-dimensional control tasks. Each curve is averaged over 10 independent experiments.

Acknowledgments We would like to thank Arnab Bhattacharyya for helpful discussions on our work and its connection to learning and testing juntas, and Jiajin Li for pointing out that variable partitioning for reinforcement learning in fact reduces the variance of its policy gradient estimator.

References

[BCP+16] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

[BU87] S.S. Barsov and Vladimir Ulyanov. Estimates of the proximity of gaussian measures. *Doklady Mathematics*, 34:462–, 01 1987.

[CL15] Chandra Chekuri and Shi Li. A note on the hardness of approximating the k-way hypergraph cut problem, 2015.

[CXY18] Karthekeyan Chandrasekaran, Chao Xu, and Xilin Yu. Hypergraph k-cut in randomized polynomial time. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '18, pages 1426–1438, Philadelphia, PA, USA, 2018. Society for Industrial and Applied Mathematics.

[DDG+17] Roee David, Irit Dinur, Elazar Goldenberg, Guy Kindler, and Igor Shinkar. Direct sum testing. *SIAM Journal on Computing*, 46(4):1336–1369, 2017.

[DWS12] Thomas Degris, Martha White, and Richard S Sutton. Off-policy actor-critic. *arXiv preprint arXiv:1205.4839*, 2012.

[Kar00] David R. Karger. Minimum cuts in near-linear time. *J. ACM*, 47(1):46–76, January 2000.

[Kos18] Ilya Kostrikov. Pytorch implementations of reinforcement learning algorithms. `https://github.com/ikostrikov/pytorch-a2c-ppo-acktr`, 2018.

[KS96] David R. Karger and Clifford Stein. A new approach to the minimum cut problem. *J. ACM*, 43(4):601–640, July 1996.

[KS09] Marek Karpinski and Warren Schudy. Linear time approximation schemes for the gale-berlekamp game and related minimization problems. In *Proceedings of the Forty-first Annual ACM Symposium on Theory of Computing*, STOC '09, pages 313–322, New York, NY, USA, 2009. ACM.

[KSL15] David Karger and Matthew S. Levine. Fast augmenting paths by random sampling from residual graphs. *SIAM Journal on Computing*, 44:320–339, 03 2015.

[KT15] Ken-ichi Kawarabayashi and Mikkel Thorup. Deterministic global minimum cut of a simple graph in near-linear time. In *Proceedings of the Forty-seventh Annual ACM Symposium on Theory of Computing*, STOC '15, pages 665–674, New York, NY, USA, 2015. ACM.

[KW96] Regina Klimmek and Frank Wagner. A simple hypergraph min cut algorithm, 1996. Technical Report B.

[LW18] Jiajin Li and Baoxiang Wang. Policy optimization with second-order advantage information. *arXiv preprint arXiv:1805.03586*, 2018.

[Man17] Pasin Manurangsi. Almost-polynomial ratio ETH-hardness of approximating densest k-subgraph. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2017, pages 954–961, New York, NY, USA, 2017. ACM.

[MBM+16] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy P Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International Conference on Machine Learning*, 2016.

[Que98] Maurice Queyranne. Minimizing symmetric submodular functions. *Mathematical Programming*, 82(1-2):3–12, 1998.

[RSW18] Aviad Rubinstein, Tselil Schramm, and S. Matthew Weinberg. Computing Exact Minimum Cuts Without Knowing the Graph. In Anna R. Karlin, editor, *9th Innovations in Theoretical Computer Science Conference (ITCS 2018)*, volume 94 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 39:1–39:16, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[RV07] R. M. Roth and K. Viswanathan. On the hardness of decoding the gale-berlekamp code. In *2007 IEEE International Symposium on Information Theory*, pages 1356–1360, June 2007.

[SB18] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[SML+15] John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015.

[SMSM00] Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, pages 1057–1063, 2000.

[SWD+17] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[Wil92] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.

[WRD+18] Cathy Wu, Aravind Rajeswaran, Yan Duan, Vikash Kumar, Alexandre M.Bayen, Sham Kakade, Igor Mordatch, and Pieter Abbeel. Variance reduction for policy gradient with action-dependent factorized baselines. In *International Conference on Learning Representations*, 2018.

A Statistical claims

Claim 27. *Assume $t, \hat{t} \geq 0$. If $t^p \leq \hat{t}^p \leq t^p + (\varepsilon/2)^p$ then $t \leq \hat{t} \leq t + \varepsilon$.*

Proof. The left-hand inequalities are immediate. For the right-hand ones we start we consider two cases. If $t \leq \varepsilon/2$, then $\hat{t}^p \leq 2(\varepsilon/2)^p \leq \varepsilon \leq t + \varepsilon$. If $t > \varepsilon/2$ then

$$\hat{t} - t \leq \frac{\hat{t}^p - t^p}{t^{p-1}} \leq \frac{(\varepsilon/2)^p}{(\varepsilon/2)^{p-1}} \leq \varepsilon. \qquad \square$$

Claim 7. *Assuming $\|F\|_{\mathbb{R},2p} \leq 1$, the value $\|F\|_{\mathbb{R},p}^p$ can be estimated within ε^p, and $\|F\|_{\mathbb{R},p}$ can be estimated within ε, from $K^p \log(1/\gamma)/\epsilon^{2p}$ queries to F in linear time with probability $1-\gamma$ for some absolute constant K.*

Proof. By Chebyshev's inequality, $\mathbb{E}[|F|^p]$ can be estimated within an additive error of $(\varepsilon/2)^p$ by averaging $(2/\varepsilon)^{2p}$ samples with probability $3/4$. The error can be improved to $1-\gamma$ by taking the median value of $O(\log 1/\gamma)$ runs. The second bound follows from Claim 27. $\square$

B Details in the experiments

The exact reinforcement learning control algorithm we used is described below. The algorithm is based on proximal policy optimization [SWD+17] and generalized advantage estimator [SML+15, DWS12] in reinforcement learning.

Algorithm 3 Policy optimization with variable partitions

Input: Total number of samples T, batch size B, partition frequency M_p, number of value iterations M_w, initial policy parameter θ, initial value and advantage parameters w and μ;
Output: Optimized policy π_θ;
for each iteration j in $[T/B]$ **do**
    Collect a batch of trajectory data $\{s_t^{(i)}, a_t^{(i)}, r_t^{(i)}\}_{i=1}^B$;
    for M_θ iterations **do**
        Update θ by one gradient descent step using proximal policy gradient with the gradient estimator Eq. (10);
    end for
    for M_w iterations **do**
        Update w and μ by minimizing $\|V^w(s_t) - R_t\|_2^2$ and $\|\hat{A} - A^\mu(s_t, a_t)\|_2^2$ in one step;
    end for
    Estimate $\hat{A}(s_t, a_t)$ using $V^w(s_t)$ by generalized advantage estimator;
    if $j \equiv 0 \pmod{M_p}$ **then**
        Define estimation $f(\boldsymbol{X}) = \mathbb{E}[D_F(\boldsymbol{X}, \overline{\boldsymbol{X}})^2]$;
        Run submodular minimization on $f(\boldsymbol{X})$;
        Assign $\boldsymbol{X}$ and $\bar{\boldsymbol{X}}$ to $a_{(1)}$ and $a_{(-1)}$ in (10), respectively;
    end if
end for

Variable Partition

The differences between our algorithm and proximal policy gradient [SWD+17] have been highlighted: Line 6 uses the estimator with partitions on the control variables. Line 12-16 find the near-optimal variable partition using submodular minimization, by Theorem 2.

We use three neural networks as function approximations: a policy network π_θ and a value network V^w as is in the baseline methods, and an advantage network A^μ solely used in the partition algorithm. The networks have the same architecture as is in previous series of works [MBM+16, SWD+17].

In our MuJoCo experiments, the tasks have been slightly modified (the physics simulator keeps intact). As the number of control variables of the original tasks are relatively low, we augment the such dimensions by letting the agent controls two independent instances of the tasks at the same time. The scores and the reinforcement signals are then the additions of the scores of the two sub-tasks. Correspondingly, we use $k = 2$ in [LW18] and our algorithms.