

# Hardness of Approximate Sperner and Applications to Envy-Free Cake Cutting

Ruiquan Gao<sup>1</sup>, Mohammad Roghani<sup>1</sup>, Aviad Rubinstein<sup>1</sup>, and Amin Saberi<sup>1</sup>

<sup>1</sup>Stanford University, {ruiquan, roghani, aviad, saberi}@stanford.edu

## Abstract

Given a so called “Sperner coloring” of a triangulation of the  $D$ -dimensional simplex, Sperner’s lemma guarantees the existence of a *rainbow simplex*, i.e. a simplex colored by all  $D+1$  colors. However, finding a rainbow simplex was the first problem to be proven PPAD-complete in Papadimitriou’s classical paper introducing the class PPAD [Pap94]. In this paper, we prove that the problem does not become easier if we relax “all  $D+1$  colors” to allow some fraction of missing colors: in fact, for any constant  $D$ , finding even a simplex with just three colors remains PPAD-complete!

Our result has an interesting application for the envy-free cake cutting from fair division. It is known that if agents value pieces of cake using general continuous functions satisfying a simple boundary condition (“a non-empty piece is better than an empty piece of cake”), there exists an envy-free allocation with connected pieces. We show that for any constant number of agents it is PPAD-complete to find an allocation—even using any constant number of possibly disconnected pieces—that makes just three agents envy-free.

Our results extend to super-constant dimension, number of agents, and number of pieces, as long as they are asymptotically bounded by any  $\log^{1-\Omega(1)}(\epsilon)$ , where  $\epsilon$  is the precision parameter (side length for Sperner and approximate envy-free for cake cutting).

# 1 Introduction

Sperner’s lemma is a fundamental result in combinatorics with important applications including topology (Brouwer’s fixed point theorem), game theory (Nash equilibrium), economics (market equilibrium), and fair division (envy-free cake cutting). The lemma is stated in terms of a *large*  $D$ -dimensional simplex, a partition of the simplex into disjoint *small*  $D$ -dimensional simplices, and a  $D + 1$ -Sperner coloring of the vertices of the small simplices which has to satisfy appropriate boundary conditions<sup>1</sup>. The lemma says that there always exists a small *rainbow simplex*, i.e. a small simplex whose vertices are colored by all  $D + 1$  colors.

Sperner’s existence result does not come with an efficient algorithm. Indeed, when the coloring is given by a circuit, finding a rainbow simplex is PPAD-complete, even for the case of  $D + 1 = 3$  colors [CD09] (for  $D + 1 = 2$  colors, a bichromatic simplex (edge) can be found easily by a binary search). The PPAD-completeness of Sperner’s lemma has been used to show analogous hardness results for all its aforementioned applications (Brouwer [Pap94], Nash [DGP09], market equilibrium [VY11; CPY17], cake-cutting [DQS12; HR23], and more [KPR+09; OPR16; GH21]).

Given the hardness of finding a small rainbow simplex, in this work we ask whether the problem becomes easier when relaxing the problem to only requiring the small simplex to have most of the colors. Our main technical contribution is a strong impossibility result for this relaxation of Sperner’s lemma:

**Theorem 1** (informal version of [Theorem 15](#)). *For any  $D > 0$ , given circuit (resp. oracle) access to a  $D$ -dimensional Sperner coloring with  $\varepsilon^D$ -side-length small simplices, finding even a trichromatic small simplex is PPAD-complete (resp. requires  $\text{poly}(1/\varepsilon)$  oracle queries).*

Note that this result is tight in the sense that as mentioned earlier, finding a bichromatic simplex is easy (using binary search). It is also interesting that even though the number of colors in the existence result grows linearly with  $D$ , the number of colors we can find algorithmically does not grow at all.

## Cake cutting

It is interesting to understand what consequences we can get from [Theorem 15](#) for PPAD-complete applications of Sperner’s lemma. In this work, we focus on the envy-free cake cutting problem from the study of fair division: how can we fairly partition and allocate a heterogeneous “cake” over the unit-interval between  $n$  agents?

Briefly (details deferred to [Section 8](#)), in the cake cutting model each agent has a value function mapping a partition<sup>2</sup> of the cake to a value in  $[0, 1]$  for each of the pieces. An allocation of cake is *envy-free* if no agent envies another agent’s allocation. A classical result (e.g. [Su99] using Sperner’s lemma) shows that an envy-free allocation always exists even if we insist on giving each agent a single contiguous (or *connected*) piece of cake. This result holds for a general class of value functions, that are only required to (i) be continuous in the location of the cuts and (ii) assign value 0 to a piece if and only if it has length 0. In particular, the value for a piece of cake may be non-monotone (i.e. less cake is sometimes better), and more generally depend on the entire partition, not just the cuts at its endpoints.

While an envy-free allocation with connected pieces always exists, it is not clear how to algorithmically find it. In order to cast this problem in a standard computational model, [DQS12] propose studying the complexity of  $\varepsilon$ -EF cake cutting, i.e. we want an allocation where no agent envies another agent’s allocation by more than an additive  $\varepsilon$ . (To make the additive guarantee meaningful we normalize the values to  $[0, 1]$  and

---

<sup>1</sup>Namely that each vertex of the large simplex has a different color, and any small-simplex-vertex on a large-simplex-facet shares a color with one of the large-simplex-vertices of that facet.

<sup>2</sup>We use the following terminology: (i) a configuration of *cuts*, is just a finite subset of  $[0, 1]$ ; (ii) the *partition* of the cake into disjoint, connected pieces as determined the cuts; (iii) the *bundling* of the pieces into disjoint subsets; and finally (iv) the *allocation* which determines which agent receives which subset.

insist that the value functions are Lipschitz.) They prove that  $\epsilon$ -EF cake cutting is indeed PPAD-complete, like the problem associated with Sperner’s lemma.

In this work we use our main technical theorem to show that  $\epsilon$ -EF cake cutting continues to be PPAD-complete even under two natural relaxations of the problem:

**Can almost everyone be almost envy-free?** Given that it is intractable to find an allocation where no agent has  $\epsilon$ -envy, it would be desirable to find an allocation where at least most agents are envy-free. Unfortunately, we show that for any constant number of agents, it is PPAD-complete to find an allocation where even 3 agents do not envy any other agents.

**Beyond connected pieces**<sup>3</sup> In most applications it is desirable to have as few cuts as possible (e.g. when the cake models a resource like a computing cluster or vacation home shared across time), but connected pieces may not be a strict requirement. Indeed, in the (important) special case of additive value functions, finding efficient algorithms with *general*, i.e. possibly disconnected, pieces has been a famous open problem for several decades. A celebrated breakthrough of [AM20] shows that for any constant number of agents, there is an efficient algorithm that finds an envy-free allocation with a constant number of cuts — this is in contrast to connected pieces where a  $\text{polylog}(1/\epsilon)$ -time algorithm was only recently given for  $n = 4$  agents [HR23], and  $n \geq 5$  agents remain an open problem. Here we show that for more general value functions, the problem becomes PPAD-complete, even with 3 agents and any constant number of cuts.

**Theorem 2** (Informal version of Theorem 20). *For any number  $k \geq 3$  of agents and cuts, it is PPAD-hard to find an  $\epsilon^k$ -EF allocation (of possibly disconnected pieces) that makes even 3 agents envy-free.*

Our PPAD-completeness result imply  $\text{poly}(1/\epsilon)$  lower bounds on the number of value queries (respectively, color queries for Sperner’s lemma). We remark also that our lower bounds for cake cutting hold for the special case where all agents have the same value function.

## 1.1 Closely related work

Our result on approximate Sperner’s lemma is most closely related to works on approximate Brouwer fixed point [HPV89; Rub18; BPR16; Rub16; FHSZ23]. In particular, [Rub16] proves that in the regime of asymptotically large dimension, it is PPAD-complete to find a small simplex with  $1 - \delta$  fraction of the colors (for a small, unspecified constant<sup>4</sup>  $\delta > 0$ ) even when the small simplex has constant side length<sup>5</sup>  $\epsilon > 0$ . This result is incomparable to our result which holds in constant dimension but requires the small simplex to be asymptotically small. In terms of the fraction of colors, we show that 3-vs- $\omega(1)$  is already hard, compared to  $(1 - \delta)n$ -vs- $n$  in [Rub16]. Our recursive construction is technically completely different from the error correcting code technique of [Rub16]; it is an interesting direction for future work to explore whether our result on approximate Sperner coloring can have interesting implications for approximate equilibria in games (main goal of [Rub16]), equilibria in other domains (e.g. markets), or even the PCP Conjecture for PPAD [BPR16].

For cake-cutting with additive valuations, the problem of envy-free cake cutting has been studied in the Robertson-Webb oracle model, where in addition to value queries, the algorithm can also ask the oracle for a

<sup>3</sup>We thank Alexandros Hollender for suggesting to us the connection between our notion of approximate Sperner’s Lemma and cake cutting with disconnected pieces.

<sup>4</sup>An earlier result of [Rub18] proved a weaker result where only the side length is a small unspecified constant; the constants in that paper were recently dramatically improved in [FHSZ23]. However, neither of those works has any non-trivial guarantee on the fraction of colors in the small simplex; this fraction is the focus of our paper.

<sup>5</sup>The domain in [Rub16] and other related works on approximate Brouwer fixed point is the hypercube rather than the simplex. While in constant dimension this makes little difference, in  $n$ -dimensions the simplex is exponentially smaller. Our comparison here is informal, possibly erring on the side of giving earlier work too much credit.

piece that agent  $i$  values at  $\alpha$ . For additive  $\epsilon$ -EF allocations, [BN22] show that cut queries can be simulated with  $\Theta(\log(1/\epsilon))$  value queries, so the models are equivalent up to these log factors. Cut queries have a natural extension to monotone value functions (see e.g. [HR23]), but it is not clear how to extend them to our model of general value functions.

In the Robertson-Webb oracle model, one can ask for exact EF allocations. For connected pieces, even in the Robertson-Webb oracle model no algorithm can find an EF allocation in finite time [Str08] (demonstrating a strong separation between the connected and general pieces). For general pieces, the celebrated breakthrough of [AM20] gives an efficient algorithm that runs in constant time (and hence a constant number of cuts) for a constant number of agents. However, the dependence on the number of agents  $n$  is poor: a six-power tower; in contrast, the best lower bound is only  $\Omega(n^2)$  [Pro09], and closing the gap is a famous open problem. Our result suggests that it would be impossible to find better algorithms by only exploiting the topology of the problem.

Our hardness on making most agents envy-free is related to a result of [BPR16] who showed, assuming the PCP Conjecture for PPAD, a similar-flavor notion of hardness for approximately fair allocation of courses to students in the A-CEEI framework. (The techniques are completely different, and they focus on the regime of asymptotically large number of agents/students.)

[HR23] study the complexity of  $\epsilon$ -EF cake cutting with 4 agents. They show that an efficient protocol with monotone value functions, and PPAD, query, and communication complexity hardness for non-monotone functions. Their notion of non-monotone functions is less general than our notion of value functions in the sense that in [HR23] the value of a piece only depends on the locations of the cuts that define it. It is an interesting open problem whether our results can be extended to this notion of value functions, and/or to communication complexity (see also [BN19] on communication complexity in cake cutting).

Many other problems in fair division are known to be complete for PPAD (e.g. [OPR16; CGMM20]), or related classes such as PPA [FG18; DFHM22; DFH22], PLS [GHH23], PPA- $k$  [FHSZ21], and likely also for  $\text{PPAD} \cap \text{PLS}$  [ABR19]; see also the recent survey of [AAB+23]. In particular, [DFHM22] extend a PPA-completeness result of fair division (“consensus halving”) with  $n$  cuts to allow a small fraction of redundant cuts, namely  $n + n^{1-\delta}$ ; in comparison, we show hardness of  $3\text{-vs-}\omega(1)$  cuts.

## 2 Overview of techniques

To prove hardness of approximate Sperner (Theorem 15) we use a combination of a recursive construction and the hard 2-dimensional instance of [CD09]. At the base of our recursion we consider a  $D = 1$ -dimensional instance, which is simply a line with two colors. We assume wlog that the line is colored with blue for all the point left of the midpoint, and red on the right.

The recursive step builds on a construction of [CD09] who showed that given a Sperner coloring in  $D = 2$  dimensions, it is PPAD-complete to find a small trichromatic simplex. We can think of their construction as extending a 1-dimensional base case to a 2-dimensional instance. We use their construction almost black box, except again for the assumption (which can be easily seen to hold in their construction) that the base case is colored with blue up to the midpoint, and red thereafter.

### The key idea

At a high level, for going from a generic  $D$  to  $D + 1$  dimensions, we consider all the points where two colors meet. Locally, consider the line segment perpendicular to the manifold between the two colors, and observe that it looks just like a shrunk version of our 1-dimensional base case (one color up to the midpoint, then a different color). So we can extend it in the  $(D + 1)$ -th dimension from a line segment to a 2-dimensional simplex by pasting a shrunk copy of [CD09]’s construction. By default, all other points with a non-trivial component in the  $(D + 1)$ -th dimension are colored with the  $(D + 1)$ -th color.

Suppose by induction that it is PPAD-hard to find a point simultaneously close to 3 colors in the  $D$ -dimensional instance. The base case is trivial as clearly with  $D + 1 = 2$  colors there are no trichromatic regions at all. For the induction step, we need to argue that it is hard to find a point in the  $(D + 1)$ -dimensional instance that is simultaneously close to the new  $(D + 1)$ -th color and two of the other  $D$  colors. However, any point close to two of the  $D$  colors is covered by a shrunk copy of the 3-color, 2-dimensional [CD09]-instance, where it is again PPAD-complete to find a trichromatic point!

### Challenge: maintaining consistency with empty pieces

The most interesting challenge in implementing the above proof plan has to do with the inherent symmetries of the cake cutting applications. Consider for example the case where we only try to make some agents envy-free (a similar challenge arises for redundant cuts): in this case, it is acceptable to completely sacrifice the happiness of one agent and give them an empty piece. Now there are  $D - 1$  configurations of  $D$  cuts that correspond to the same allocation of  $D + 1$  cake pieces with one of them being empty. For the cake cutting application, we would like to ensure that the agents' values for all those allocations to are identical.

Geometrically, “ $i$ -th piece is empty” corresponds to a tight constraint on the location of the cuts, or equivalently a facet of the simplex. So in order to correspond to a feasible value function, the Sperner coloring has to satisfy an unusual symmetry constraint:

For every  $d < D$ , the coloring on every  $d$ -dimensional the facets is identical, up to a permutation of the colors.

Notice that this issue comes up specifically because of the relaxed notion of envy-free with respect to some agents that we study in this paper: in contrast, for every-agent-envy-free cake cutting with connected pieces, we can assume wlog that every piece is nonempty (equivalently, there is no rainbow simplex on the boundary) — for any agent that receives it will envy other agents.

Recall that the  $D - 1$  dimensional construction is constructed from shrunk copies of the 2-dimensional instance. When we recurse to add more dimensions, the copies have to shrink exponentially, resulting in different *shrinking factors*. Thus our Sperner construction has different shrinking factors for different facets — failing to satisfy the symmetry condition.

To overcome this, we modify our  $D$ -dimensional construction to first have a copy of the  $D - 1$ -dimensional construction on each facet. Then in the regions near the boundary we interpolate between this original copy and the copy with smaller shrinking factors.

## 3 Preliminaries

We use  $[n]$  to denote the set  $\{1, \dots, n\}$ . We use  $\mathcal{Q}_n$  to denote the integer set  $\{0, \dots, 2^n - 1\}$  that can be encoded by  $n$  bits. We use  $(x)_+$  to denote  $\max\{0, x\}$ . We use  $(x)_-$  to denote  $\min\{1, x\}$ . Further, we use  $(x)_{[0,1]}$  to denote  $((x)_+)_-$ . For any vector  $\mathbf{x}$  and any  $l, r \in \mathbb{Z}$ , we use  $\mathbf{x}_{l:r}$  to denote  $\mathbf{x}$  restricted to indices in  $[l, r]$ , i.e.,  $(x_l, \dots, x_r)$ . For any vector  $\mathbf{x}$  and any  $i \in \mathbb{Z}$ , we use  $\mathbf{x}_{-i}$  to denote  $\mathbf{x}$  restricted to indices not equalling  $i$ , i.e.,  $(\mathbf{x}_{1:i-1}, \mathbf{x}_{i+1:r})$ .

**The Complexity Class:** TFNP. Search problems are defined via relations  $R \subseteq \{0, 1\}^* \times \{0, 1\}^*$ , where the goal is to find a string  $\mathbf{y}$  for the input  $\mathbf{x}$  such that  $(\mathbf{x}, \mathbf{y}) \in R$ , or to claim no such  $\mathbf{y}$  exists. The class FNP consists of all such search problems in which  $R$  is polynomial-time computable, which means that whether any  $(\mathbf{x}, \mathbf{y}) \in R$  can be decided in polynomial time, and polynomially balanced, which means that there exists a polynomial  $p(n)$  such that  $(\mathbf{x}, \mathbf{y}) \in R$  only if  $|\mathbf{y}| \leq p(|\mathbf{x}|)$ . Here,  $|\mathbf{x}|$  is defined as the number of bits in

string  $\mathbf{x}$ . The class TFNP consists of all FNP problems whose relation  $R$  is total, i.e., for any input  $\mathbf{x}$ , there always exists a string  $\mathbf{y}$  such that  $(\mathbf{x}, \mathbf{y}) \in R$ .

The polynomial-time reduction between two TFNP problems  $R, R'$  is defined by two polynomial-time computable functions  $f : \{0, 1\}^* \rightarrow \{0, 1\}^*$  and  $g : \{0, 1\}^* \times \{0, 1\}^* \rightarrow \{0, 1\}^*$  such that for any instance  $\mathbf{x}$  of  $R$ , we can always reduce it to the instance  $f(\mathbf{x})$  (of  $R'$ ) and then recover a solution from any output  $\mathbf{y}$  of the instance of  $R'$  by  $g(\mathbf{x}, \mathbf{y})$ . That is, for any input  $\mathbf{x}$  of  $R$  and any output  $\mathbf{y}$  of  $R'$ ,

$$(f(\mathbf{x}), \mathbf{y}) \in R' \Rightarrow (\mathbf{x}, g(\mathbf{x}, \mathbf{y})) \in R.$$

**The Complexity Class: PPAD.** The class PPAD consists of all TFNP problems that are polynomial-time reducible to the following END-OF-LINE problem.

**Definition 3 (END-OF-LINE).** In END-OF-LINE, we are given a predecessor function  $P : Q_n \rightarrow Q_n$  and a successor function  $S : Q_n \rightarrow Q_n$  such that  $P(0^n) = 0^n \neq S(0^n)$ , where both  $P$  and  $S$  are given by a polynomial-size circuit. The goal of END-OF-LINE is to find  $x \in Q_n$  such that we have either  $P(S(\mathbf{x})) \neq \mathbf{x}$  or  $S(P(\mathbf{x})) \neq \mathbf{x} \neq 0^n$ .

This class, PPAD, captures the complexity of the following  $k$ D-SPERNER problem, whose totality is guaranteed by the famous Sperner's lemma [Spe28].

**Definition 4.** The standard  $k$ -simplex is  $\Delta^k = \{\mathbf{x} \in [0, 1]^{k+1} : \sum_{i=1}^{k+1} x_i = 1\}$ .

**Definition 5.** The discrete  $k$ -simplex with triangulation side-length of  $2^{-n}$  is defined as the following subset of the standard  $k$ -simplex  $\Delta_n^k$ :  $\Delta_n^k = \{\mathbf{x} \in \Delta^k : \forall i \in [k+1], (2^n - 1) \cdot x_i \in \mathbb{Z}\}$ .

**Definition 6.** The array of non-trivial indices  $\mathcal{I}_{>0}(\mathbf{x})$  of  $\mathbf{x}$  is given by indices  $i$  such that  $x_i > 0$  in an increasing order.

**Definition 7 (kD-SPERNER).** In  $k$ D-SPERNER, we are given function  $C : \Delta_n^k \rightarrow [k+1]$  satisfying  $C(\mathbf{x}) \in \mathcal{I}_{>0}(\mathbf{x})$  for any  $\mathbf{x} \in \Delta_n^k$ . The goal of  $k$ D-SPERNER is to find  $k+1$  points  $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(k+1)} \in \Delta_n^k$  such that

1. **the three points are close to each other**, i.e.,  $\forall i, j \in [k+1], \|\mathbf{x}^{(i)} - \mathbf{x}^{(j)}\|_\infty \leq 2^{-n}$ ; and
2. **the induced simplex is rainbow**, i.e.,  $|\{C(\mathbf{x}^{(i)}) : i \in [k+1]\}| = k+1$ .

**Theorem 8 ([HPV89; Pap94; CD09]).** For any  $k \geq 2$ ,  $k$ D-SPERNER is PPAD-complete (if the coloring  $C$  is given by a polynomial-size circuit) and requires a query complexity of  $2^{\Omega(n)}$  (if the coloring  $C$  is given by a black box).

## 4 Approximate High-dimensional Sperner: Formulation and Statement of Result

In this section, we formally define the relaxed computational problem of Sperner's lemma, and its variants that we will use for the applications on envy-free cake cutting. In particular, we state our main technical result [Theorem 15](#), deferring the proof to [Sections 6](#) and [7](#).

Here, we use the term *approximate Sperner's lemma* for the following relaxed version of the original Sperner's lemma [Spe28]: in any triangulation of a large  $k$ -dimensional simplex with a  $k+1$ -Sperner coloring that satisfies appropriate boundary condition, there always exists a small triangle (a.k.a., 2-simplex) whose three vertices are colored differently (because Sperner's lemma states that there exists a small  $k$ -simplex whose  $k+1$  vertices are colored by all  $k+1$  colors). Formally, we define the following 3-out-of- $k+1$  Approximate SPERNER problem.

**Definition 9** (3-out-of- $k+1$  Approximate SPERNER). In 3-out-of- $k+1$  Approximate SPERNER (with a triangulation side-length of  $2^{-n}$ ), we are given a function  $C : \Delta_n^k \rightarrow [k+1]$  satisfying  $C(\mathbf{x}) \in \mathcal{I}_{>0}(\mathbf{x})$  for any  $\mathbf{x} \in \Delta_n^k$ . The goal of 3-out-of- $k+1$  Approximate SPERNER is to find three points  $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3 \in \Delta_n^k$  such that

1. **the three points are close to each other**, i.e.,  $\forall i, j \in [3], \|\mathbf{x}_i - \mathbf{x}_j\|_\infty \leq 2^{-n}$ ; and
2. **the induced triangle is trichromatic**, i.e.,  $|\{C(\mathbf{x}_i) : i \in [3]\}| = 3$ .

**Remark 10.** When  $k = 2$ , the 3-out-of- $k+1$  Approximate SPERNER problem is simply equivalent to the  $kD$ -SPERNER problem.

To apply the hardness result of 3-out-of- $k+1$  Approximate SPERNER to the envy-free cake-cutting problem, we further consider the special case where the coloring defined by  $C$  satisfies the following symmetry constraints on the boundary of the simplex. First, we define the symmetry that naturally arises from cake-cutting valuations. For example, suppose we cut the cakes using two cuts in two different ways: with cut locations  $(0.1, 0.1)$  and  $(0.1, 1)$ . Both partitions of the cake have the same two non-trivial pieces:  $[0, 0.1]$  and  $[0.1, 1]$ . Thus we expect that the preference for each agent between those pieces is the same regardless of the representation on the simplex. This inspires us to define a symmetry constraint on cuts that have the same set of non-zero pieces.

**Definition 11.** Indexing  $\text{index}(\mathbf{a}, v)$  of  $v$  in an array  $\mathbf{a}$  is defined as the index  $i$  such that  $a_i = v$ .

**Definition 12** (symmetric points in a coloring). We say  $\mathbf{x}$  and  $\mathbf{y}$  are symmetric in a coloring  $C$  if

- the number of non-trivial entries in  $\mathbf{x}$  and  $\mathbf{y}$  are the same, i.e.,  $|\mathcal{I}_{>0}(\mathbf{x})| = |\mathcal{I}_{>0}(\mathbf{y})|$ .
- the indexing of  $C$  are the same for the arrays of non-trivial indices of  $\mathbf{x}$  and  $\mathbf{y}$ , i.e.,  $\text{index}(\mathcal{I}_{>0}(\mathbf{x}), C(\mathbf{x})) = \text{index}(\mathcal{I}_{>0}(\mathbf{y}), C(\mathbf{y}))$ .

We use  $\mathbf{x} \sim_C \mathbf{y}$  to denote this symmetry.

Because the two conditions for symmetry are both equations, which satisfy reflexivity, symmetry, and transitivity, this symmetry is a valid equivalence relation.

**Fact 13.** For any coloring  $C$ , the symmetry of points in coloring  $C$  is an equivalence relation, i.e., a binary relation satisfying reflexive ( $\mathbf{x} \sim_C \mathbf{x}$ ), symmetric ( $\mathbf{x} \sim_C \mathbf{y} \rightarrow \mathbf{y} \sim_C \mathbf{x}$ ) and transitive ( $\mathbf{x} \sim_C \mathbf{y}$  and  $\mathbf{y} \sim_C \mathbf{z} \rightarrow \mathbf{x} \sim_C \mathbf{z}$ ).

In the symmetric version of the 3-out-of- $k+1$  Approximate SPERNER problem, the input guarantees that symmetric points on the boundary are symmetric in the input coloring. We say two points on the boundary are symmetric to each other, if the resulting vectors are the same after we remove the zero entries in them.

**Definition 14** (3-out-of- $k+1$  Approximate Symmetric SPERNER). 3-out-of- $k+1$  Approximate Symmetric SPERNER is a special case of 3-out-of- $k+1$  Approximate SPERNER, where the given circuit  $C$  further satisfies the following property: for any  $\mathbf{x} \in \Delta_n^k$  and any  $i, j \in [k+1]$ ,  $(\mathbf{x}_{1:i-1}, 0, \mathbf{x}_{i:k}) \sim_C (\mathbf{x}_{1:j-1}, 0, \mathbf{x}_{j:k})$ .

Our main technical result is the PPAD-hardness of this 3-out-of- $k+1$  Approximate Symmetric SPERNER problem. Because this problem is clearly a member of the class PPAD, its complexity is PPAD-complete.

**Theorem 15.** For and any  $k = \text{poly}(n)$ , 3-out-of- $k+1$  Approximate Symmetric SPERNER with a triangulation side-length of  $2^{-4kn}$  is PPAD-complete. Further, for any  $k \geq 2$ , it requires a query complexity of  $2^{\Omega(n)} / \text{poly}(n, k)$ .

## 5 Cake Cutting: Formal Model and Statement of Result

### Cuts, pieces and allocations

***k*-cuts:** In line with the classic model of cake cutting, we use a line segment  $[0, 1]$  to represent the “cake”. We let  $k$  be the number of the cuts. A *k*-cut can be represented as a vector  $\mathbf{x} = (x_1, x_2, \dots, x_k)$  with  $k$  dimensions such that  $0 \leq x_1 \leq x_2 \leq \dots \leq x_k \leq 1$ . Dimension  $i$  shows where the  $i$ -th cut is located on  $[0, 1]$ .

We only define preferences with respect to partitions that use exactly  $k$  cuts, for some parameter  $k$  (note that we allow  $k$  to be larger than the number of agents). Partitions corresponding to fewer than  $k$  cuts can easily<sup>6</sup> be incorporated in this model by adding trivial cuts (equivalently, empty pieces). We do not allow more than  $k$  cuts in our model. Note that without an upper bound on the number of cuts, one could partition the cake into infinitesimal pieces and assign them to agents at random; by continuity of the valuations this would trivially be approximately envy-free, but eating infinitely many disconnected infinitesimal pieces of cake can be unsatisfactory in applications.

**Pieces of cake:** A *piece of cake* is an open interval that is a subset of  $[0, 1]$ . So a  $k$ -cut partitions the cake into  $k + 1$  (possibly empty) pieces<sup>7</sup>. For a  $k$ -cut  $\mathbf{x}$ , we use  $X = (X_0, X_1, \dots, X_k)$  (capitalizing the character representing the  $k$ -cut) to denote the corresponding pieces of the cake where  $X_0$  and  $X_k$  are the leftmost and the rightmost pieces of the cake, respectively.

**Equivalent  $k$ -cuts:** We say that two  $k$ -cuts  $\mathbf{x}$  and  $\mathbf{y}$  are *equivalent* if they induce the same pieces of cake. (For example,  $\mathbf{x} = (1/3, 1/3)$  and  $\mathbf{y} = (0, 1/3)$  both induce the pieces  $(0, 1/3), (1/3, 1), \emptyset$ .) We use  $[\mathbf{x}]$  to denote the equivalence class of  $\mathbf{x}$ .

**Allocation:** An *allocation* is an assignment of pieces to a set of agents  $\{1, 2, \dots, p\}$  (possibly multiple pieces to the same agent).

### Utilities and preferences

**Utility functions:** Each agent  $d$  has a *utility function*  $u_d$  that takes a piece of cake and the entire partition of the cake into an (unordered) set of pieces, and maps them to a real value in  $[0, 1]$ . Equivalently, and consistent with the notation we will use,  $u_d$  maps a piece  $X_i$  and an equivalence class of  $k$ -cuts  $[\mathbf{x}]$  to  $u_d([\mathbf{x}], X_i) \in [0, 1]$ . We assume that our utility functions satisfy two conditions commonly found in cake-cutting literature. The first condition is Lipschitz continuity which lets us to discretize the problem. The second condition is Nonnegativity condition which is the requirement for the existence of a valid solution for cake-cutting. We formally define these two conditions as follows:

- **Lipschitz condition:** Let  $\mathbf{x}$  and  $\mathbf{y}$  be two  $k$ -cuts. For any player  $d$ , we have  $|u_d([\mathbf{x}], X_i) - u_d([\mathbf{y}], Y_i)| \leq L \cdot \|\mathbf{x} - \mathbf{y}\|_1$  for all  $i \in \{0, \dots, k\}$ , where  $L$  is the Lipschitz constant.
- **Nonnegativity condition:** Let  $k$ -cuts  $\mathbf{x}$  be a  $k$ -cut. For any player  $d$ , we have  $u_d([\mathbf{x}], X_i) > 0$  if  $X_i \neq \emptyset$ , and  $u_d([\mathbf{x}], X_i) = 0$  otherwise.

<sup>6</sup>This is easy to *model*. But for the *proof*, the boundary conditions imposed by those empty pieces are actually one of our main sources of difficulty!

<sup>7</sup>... and their  $k + 2$  endpoints (since we treat pieces as open intervals).



**Bundles of cake pieces:** Given a partition of the cake into (connected) pieces, a *bundling*  $\mathcal{B} = \{B_1, B_2, \dots, B_p\}$  partitions the set of cake pieces into *bundles*, or subsets of pieces. The utility of a bundle of pieces for one agent is the sum of the utility of the pieces belonging to the bundle. Formally, if  $B = \{X_{i_1}, X_{i_2}, \dots, X_{i_r}\}$  is a bundle, we have  $u_d([\mathbf{x}], B) = \sum_{j=1}^r u_d([\mathbf{x}], X_{i_j})$ .

**Preferences:** Given a bundling  $\mathcal{B} = \{B_1, B_2, \dots, B_p\}$ , we say agent  $d$  (Weakly) *prefers* bundle  $B_i$  if  $u_d([\mathbf{x}], B_i) \geq u_d([\mathbf{x}], B_j)$  for all  $j$ .

## 5.1 Envy-free cake cuts

**Definition 16** ( $\epsilon$ -Approximate Envy-Free Cake Cut). *Given a  $k$ -cut  $\mathbf{x}$ , a bundling  $\mathcal{B}$  of the induced cake pieces, an assignment  $\pi : [p] \rightarrow [p]$  of the bundles to agents, and  $\epsilon > 0$  we say that  $(\mathbf{x}, \mathcal{B}, \pi)$  is  $\epsilon$ -approximate envy-free if  $u_d([\mathbf{x}], B_{\pi(d)}) + \epsilon \geq u_d([\mathbf{x}], B_i)$  for every agent  $d$  and any bundle  $B_i$ . When  $\epsilon = 0$ , we simply say that  $(\mathbf{x}, \mathcal{B}, \pi)$  is envy-free.*

Put in this language, prior work showed that with connected pieces, an envy-free cake cut always exists, but finding one is PPAD-complete:

**Theorem 17** (Stromquist [Str80] and Su [Su99]). *There exists an envy-free cake cut for  $k + 1$  agents with  $k$  cuts.*

**Theorem 18** (Deng, Qi, and Saberi [DQS12]). *Finding  $\epsilon$ -approximate envy-free cake cut for  $k + 1$  agents using  $k$  cuts is PPAD-complete.*

We consider two different settings that are less restrictive compared to the result of Theorem 18 and we prove that even if we allow both relaxations, the problem is still PPAD-complete. We study the following relaxations of the cake-cutting:

- **Making almost every agent almost envy-free:** similar to the setting in [DQS12], consider  $k$ -cut and  $k + 1$  agents, with the objective of allocating each agent a single continuous piece obtained by  $k$  cuts. Instead of seeking an envy-free  $k$ -cut, one might ask whether it is possible to find a  $k$ -cut where the majority of the agents do not envy anyone. Specifically, if  $k + 1 > 3$ , can we find a  $k$ -cut where at least three agents do not envy anyone? We provide a negative answer to this question and demonstrate that this relaxed version of the cake-cutting problem is also PPAD-complete.
- **Three agents with multiple pieces:** given the current results, a natural question arises: if we allow agents to have a bundle of pieces of the cake, is the problem still hard? It is worth noting that this question is computationally easier compared to the scenario where we allocate only one piece to each agent, as we have the option to allocate empty pieces to agents. Suppose that we have three agents and  $k + 1 > 3$  pieces of cake. Essentially, we cut the cake into  $k + 1$  pieces and bundle these pieces into three bundles. The goal is to determine whether such a bundling exists where we can allocate to each agent a bundle of pieces such that agents do not envy each other. Similarly, we demonstrate that this problem is PPAD-complete.

More specifically, we combine both relaxations. We assume that we have  $p$  agents and  $k$  cuts such that  $p \leq k + 1$ . We bundle the  $k + 1$  pieces to  $p$  bundles and allocate them to the agents, and our objective is to find an approximate envy-free cake cut where at least three agents do not envy others.

**Definition 19** ( $\epsilon$ -Approximate  $p'$ -out-of- $p$ -Envy-Free Cake Cut). *Given a  $k$ -cut  $\mathbf{x}$ , a bundling  $\mathcal{B}$  of the induced cake pieces, an assignment  $\pi : [p] \rightarrow [p]$  of the bundles to agents,  $\epsilon > 0$ , and  $p' < p$  we say that  $(\mathbf{x}, \mathcal{B}, \pi)$  is  $\epsilon$ -approximate  $p'$ -out-of- $p$ -envy-free if there exists a subset  $S \subseteq [p]$  of  $p'$  agents, such that for every  $d \in S$ ,  $u_d([\mathbf{x}], B_{\pi(d)}) + \epsilon \geq u_d([\mathbf{x}], B_i)$  for every bundle  $B_i$ .*

We can now formally state our main result for cake cutting.

**Theorem 20.** *For any constants  $3 \leq p \leq k + 1$  and  $\epsilon$  such that  $k < \log^{1-\delta}(1/\epsilon)$  for some constant  $\delta > 0$ , the problem of finding an  $\epsilon$ -approximate 3-out-of- $p$ -envy-free cake cut with  $k$  cuts is PPAD-complete. If, instead, the algorithm has black-box access to a value oracle, it requires a query complexity of  $(1/\epsilon)^{\Omega(1/k)}/\text{polylog}(1/\epsilon)$ .*

We defer the proofs and technical details to [Section 8](#).

## 6 Warmup: PPAD-Hardness of Approximate (Unconstrained) Sperner

In this section, we prove that 3-out-of- $k+1$  Approximate SPERNER (without symmetry constraints) is PPAD-complete. This is a warm-up proof for the PPAD-completeness of the 3-out-of- $k+1$  Approximate Symmetric SPERNER problem, whose hard instances and proof of PPAD-completeness will be presented in the next section. For convenience, we will not consider the query complexity in this warm-up section.

**Theorem 21.** *For any constant  $\delta > 0$  and any  $k = \text{poly}(n)$ , 3-out-of- $k+1$  Approximate SPERNER is PPAD-complete. In particular, there is a reduction in  $\text{poly}(n, k)$  time that reduces a 2D-SPERNER instance with triangulation side-length of  $2^n$  to a 3-out-of- $k+1$  Approximate SPERNER instance with triangulation side-length of  $2^{-3kn}$ .*

Because the 3-out-of- $k+1$  Approximate SPERNER problem is a relaxation of  $k$ D-SPERNER problem, which is in PPAD. The PPAD membership of 3-out-of- $k+1$  Approximate SPERNER problem is clear. Our focus of this section is to prove its PPAD-hardness.

We will now begin describing our formal construction (still for the warm-up result). To help the reader keep track of the notation, both here and the main construction in the next section, we provide a summary in [Table 1](#).

We will construct a chain of PPAD-hard instances  $C^{(2)}, C^{(3)}, \dots$  to prove this warm-up technical result. We will present the first instance of the chain  $C^{(2)}$  in [Subsection 6.1](#). Later, this  $C^{(2)}$  will also serve as the base PPAD-hard 2D-SPERNER instance  $C$  (i.e., we will use the notation  $C$  when referring to the base instance), and we will present in [Subsection 6.3](#) a recursive way that constructs  $C^{(k+1)}$  by combining  $C^{(k)}$  with one copy of this base 2D-SPERNER instance  $C$ . Towards this end, in [Subsection 6.2](#) we introduce an intermediate function, which we call the *coordinate converter*; this function projects points in a standard 2-simplex to real values in  $[0, 1]$  (the *converted coordinate*). When we move from  $C^{(k)}$  to  $C^{(k+1)}$  we have a new coordinate; we embed a copy of the base instance  $C$  on the plane spanned by the new coordinate and the converted coordinate.

Our instances will be constructed in continuous space instead of discrete space. That is, our 3-out-of- $k+1$  Approximate SPERNER instance is a function  $C^{(k)} : \Delta^k \rightarrow [k+1]$  satisfying the following hardness result. For simplicity, we define  $\epsilon := 2^{-n}$  and will use  $\epsilon$  and  $2^{-n}$  interchangeably.

**Theorem 22.** *There exists a chain of functions  $\{C^{(k)} : \Delta^k \rightarrow [k+1]\}_{k \geq 2}$  that can be computed in  $\text{poly}(|\mathbf{x}|)$  time for each  $\mathbf{x} \in \Delta^k$ , where  $|\mathbf{x}|$  is the bit complexity of  $\mathbf{x}$  and satisfies the following property. For any  $k \geq 2$ , it is PPAD-hard to find three points  $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \mathbf{x}^{(3)} \in \Delta^k$  such that*

- **they are close enough to each other:** for any  $i, j \in [3]$ ,  $\|\mathbf{x}^{(i)} - \mathbf{x}^{(j)}\|_\infty \leq 2^{-3kn} = \epsilon^{3k}$ ,
- **they induce a trichromatic triangle:**  $|\{C^{(k)}(\mathbf{x}^{(i)}) : i \in [3]\}| = 3$ .

### 6.1 Hard instances with two dimensions

In this subsection, we will present a family of continuous 2D-SPERNER instances that will serve both as the base PPAD-complete instance we will embed into this chain of 3-out-of- $k+1$  Approximate SPERNER

Table 1: Table of notation for Sections 6 and 7

Notation	Description	References
$\mathbf{x}$	vectors (also, as $\mathbf{x}'$ , $\mathbf{x}^{(1)}$ , $\mathbf{x}^{(2)}$ , ...)	all sections
$x_i$	coordinates in vectors (also, as $x'_i$ , $x_i^{(1)}$ , $x_i^{(2)}$ , ...)	all sections
$\Delta^k$	standard $k$ -simplex	Definition 4
$\Delta_n^k$	discrete $k$ -simplex with a triangulation side-length of $2^{-n}$	Definition 5
$C_{\text{rect}}(\cdot)$	2D-rectSPERNER instances	Definition 23
$C(\cdot)$	base 2D-SPERNER instances	Algorithm 1
$C^{(k)}(\cdot)$	3-out-of- $k+1$ Approximate SPERNER instances	Algorithms 1 and 2
$C_{\text{sym}}^{(k)}(\cdot)$	3-out-of- $k+1$ Approximate Symmetric SPERNER instances	Algorithm 4
$\alpha(\cdot)$	shrinking factor	Eq. (11)
$\text{rel}^{nn}(\cdot)$	coordinate converter	Eq. (5)
$\text{rel}^{nn,\alpha}(\cdot)$	coordinate converter for symmetry	Eq. (14)
$d(\cdot, \cdot)$	a metric to define $\text{rel}^{nn}$	Eq. (1)
$d^\alpha(\cdot, \cdot)$	an asymmetric quasimetric to define $\text{rel}^{nn,\alpha}$	Eq. (12)
$nn(\cdot)$	nearest neighbor	Eq. (4)
$nn^\alpha(\cdot)$	nearest neighbor w.r.t. $\alpha$	Eq. (13)
$\mathcal{N}(\cdot)$	neighborhood of size $\varepsilon \times \varepsilon$	Definition 36
$C_{nn}(\cdot)$	neighboring color (i.e., the color of the nearest neighbor)	Eq. (3)
$\hat{C}_{nn}(\cdot)$	the modified neighboring color	Eq. (7)
$C_{nn}^\alpha(\cdot)$	the neighboring color w.r.t. $\alpha$	Eq. (13)
$\hat{C}_{nn}^\alpha(\cdot)$	the modified neighboring color w.r.t. $\alpha$	Eq. (20)
$\sim_c$	symmetry in colorings (binary relation)	Definition 12
$\sim_{\text{rel}}$	equivalence between converted coordinates (binary relation)	Definition 31
$\mathbf{P}(\cdot)$	the projection step from $\Delta^k$ to $\Delta^{k-1}$ (for any $k$ )	Definition 29
$\mathbf{P}^{(\ell)}(\cdot)$	performing $\ell$ projection steps	Definition 40
$i^*(\cdot)$	the index of the first non-zero entry	Eq. (8)
$\mathbf{y}^{(i)}(\cdot)$	the $i$ -th intermediate projection (Subsections 6.3 and 7.3 only)	Algorithms 2 and 4
$\mathbf{c}^{(i)}(\cdot)$	the $i$ -th intermediate palette (Subsections 6.3 and 7.3 only)	Algorithms 2 and 4

instances and as the second instance  $C^{(2)}$  in the chain. Note that for  $k = 2$ , the 3-out-of- $k+1$  Approximate SPERNER problem is equivalent to the 2D-SPERNER problem (Remark 10).

**The instances.** This family of instances is constructed based on the hard instances of Chen and Deng [CD09]. They first construct a core rectangle which has a boundary condition similar to 2D-SPERNER, and show that it is PPAD-hard to find a small trichromatic triangle in the core rectangle. In particular, their paper can easily imply that the following rectangular version of the 2D-SPERNER problem<sup>8</sup> is PPAD-complete.

**Definition 23** (2D-rectSPERNER). Let  $Q_n^2 = \{0, 1, \dots, 2^n - 1\}^2$  denote the 2D grid with side-length of  $2^n$  on each dimension. In 2D-rectSPERNER, we are given coloring  $C : Q_n^2 \rightarrow [3]$  satisfying: (1)  $C(x, 0) \in \{1, 2\}$  and  $C(x, 2^n - 1) = 3$  for any  $x \in Q_n$ ; and (2)  $C(0, 0) = 1$ ,  $C(2^n - 1, 0) = 2$ , and  $C(0, y) = C(2^n - 1, y) = 3$  for any  $y \in Q_n \setminus \{0\}$ . The goal of 2D-rectSPERNER is to find  $\mathbf{x}^* \in (Q_n \setminus \{2^n - 1\})^2$  such that  $|\{C(\mathbf{x}) : x_i \in \{x_i^*, x_i^* + 1\}\}| = 3$ .

**Theorem 24** ([CD09]). 2D-rectSPERNER is PPAD-complete.

<sup>8</sup>In the original paper of Chen and Deng [CD09], they name this rectangular version as 2D-BROUWER.

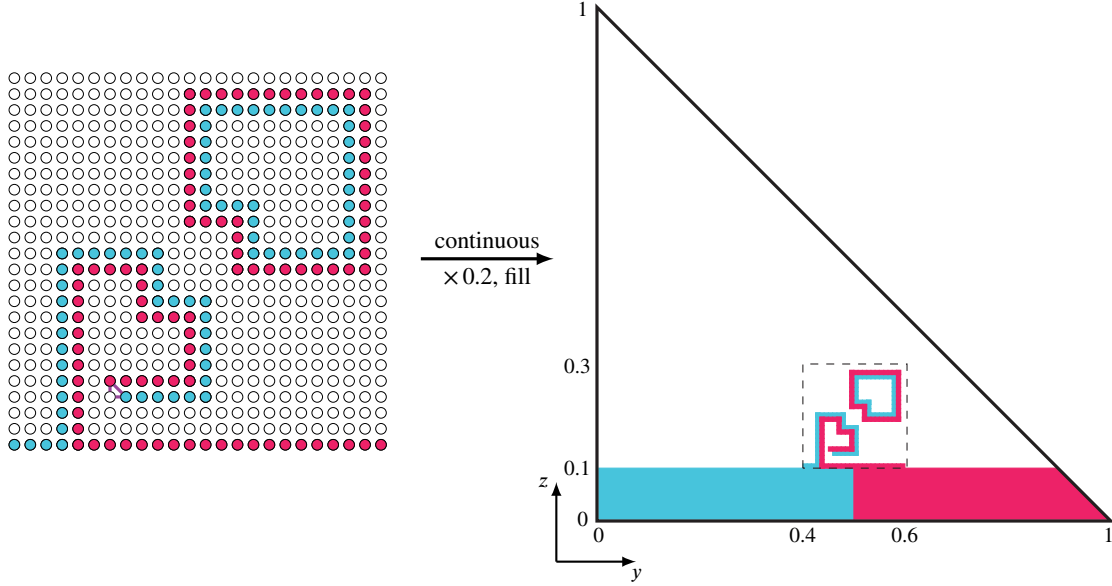


Figure 1: An example of the core rectangle in the constructions of Chen and Deng [CD09] (left) and its corresponding triangular continuous variant (right) we will consider as the base instance  $C$ . We present the triangular continuous instance by its projection onto the plane with the first coordinate  $x_1 = 0$ . The dashed square presents the *core region*. We turn the discrete instances into continuous instances by mapping each vertex in the discrete instances to a small square in the continuous instances. Then, we scale the continuous instance, put it inside a triangle, and fill the remaining part of the triangle to ensure certain boundary conditions that avoid creating spurious solutions (Facts 25 and 26 and Lemma 27).

The way Chen and Deng [CD09] use this core rectangle is embedding it into a larger (triangular) 2D-SPERNER instance, maintaining the PPAD-hardness by filling the colors outside this rectangle without creating spurious trichromatic triangles. In this paper, we will continue to use this idea but with some slight modifications. The first modification is that we turn the core rectangle into a continuous one, by mapping each point on the discrete 2D grid to a small square on a continuous 2D grid. Formally, we map each point  $(x, y)$  to the square  $[\varepsilon x, \varepsilon(x+1)) \times [\varepsilon y, \varepsilon(y+1))$ . The second modification is that we embed this core rectangle in a small region inside the 2-simplex and color for the remainder of the 2D-SPERNER instance, ensuring that (1) the bottom 1-simplex (points  $\mathbf{x} \in \Delta^2$  with the third coordinate  $x_3 = 0$ ) is equally colored by 1 and 2; (2) the other two boundaries of the 2-simplex (other than the base 1-simplex) are colored symmetrically, i.e.  $C(\mathbf{x}) = 3$  iff  $x_3 > 0.1$ . Algorithm 1 and Figure 1 conclude our embedding of the core rectangle inside a 2-simplex. We will call this part of the 2-simplex the *core region*. Facts 25 and 26 conclude the aforementioned boundary conditions. Lemma 27 establishes the computational complexity of the base instances.

**Fact 25.** For any  $y \in [0, 1]$ , the base instance  $C$  guarantees that  $C(1 - y, y, 0) = 1 + \mathbb{1}(y > 0.5)$ .

**Fact 26.** For any  $z \in [0, 1]$ , the base instance  $C$  guarantees that  $C(1 - z, 0, z) = 1 + 2 \cdot \mathbb{1}(z > 0.1)$  and  $C(0, 1 - z, z) = 2 + \mathbb{1}(z > 0.1)$ .

**Lemma 27.** Let  $\varepsilon = 2^{-n}$ . It is PPAD-hard (if  $C$  is given by a polynomial-size circuit) to find three points  $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \mathbf{x}^{(3)} \in \Delta^2$  for the 2D-SPERNER instances generated by Algorithm 1 such that

- they are close enough to each other:  $\forall i, j \in [3], \|\mathbf{x}^{(i)} - \mathbf{x}^{(j)}\|_\infty \leq \varepsilon$ ; and
- they induce a rainbow triangle:  $|\{C(\mathbf{x}^{(i)}) : i \in [3]\}| = 3$ .

---

**Algorithm 1:** Base instance  $C$ 

---

**Input** : a vector  $\mathbf{x} \in \Delta^2$  and a 2D-rectSPERNER instance  $C_{\text{rect}}$  with a side-length of  $2^{-n+3}$   
**Output** : a color  $c \in [3]$

```
1 if  $x_3 \leq 0.1$  then
2   | if  $x_2 \leq 0.5$  then
3   |   | return 1
4   |   else
5   |   | return 2 // coloring on the boundaries
6  $\varepsilon \leftarrow 2^{-n}$ 
7 if  $0.4 \leq x_2 < 0.6$  and  $x_3 < 0.3$  then
8   | return  $C_{\text{rect}}(\lfloor (1.6\varepsilon)^{-1} \cdot (x_2 - 0.4) \rfloor, \lfloor (1.6\varepsilon)^{-1} \cdot (x_3 - 0.1) \rfloor)$  // coloring in the core region
9 return 3 // default coloring
```

---

*Proof.* Suppose that  $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \mathbf{x}^{(3)}$  form a solution of the 2D-SPERNER the instance  $C$  generated by [Algorithm 1](#) using the 2D-rectSPERNER instance  $C_{\text{rect}}$ . Let  $\mathcal{D} = \{(x, y, z) \in \Delta^2 : y \in 0.5 \pm 0.1, z \in 0.2 \pm 0.1\}$  denote the core region. Because of the following [Fact 28](#), we have  $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \mathbf{x}^{(3)} \in \mathcal{D}$ . The technical proof is deferred to [Appendix A.1](#). An intuitive proof can be found in [Figure 1](#).

**Fact 28.** Suppose  $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \mathbf{x}^{(3)}$  form a solution of  $C$ , then we have  $x_2^{(j)} \in [0.4, 0.6)$  and  $x_3^{(j)} \in [0.1, 0.3)$  for any  $j \in [3]$ .

To complete the proof of [Lemma 27](#), w.l.o.g., we assume that  $C(\mathbf{x}^{(i)}) = i$  for each  $i \in [3]$ . For each  $i, j \in [3]$ , because  $\|\mathbf{x}^{(i)} - \mathbf{x}^{(j)}\|_\infty \leq \varepsilon$ , we have

$$\left| \left\lfloor (1.6\varepsilon)^{-1} \cdot (x_2^{(i)} - 0.4) \right\rfloor - \left\lfloor (1.6\varepsilon)^{-1} \cdot (x_2^{(j)} - 0.4) \right\rfloor \right| \leq 1, \quad \text{and}$$
$$\left| \left\lfloor (1.6\varepsilon)^{-1} \cdot (x_3^{(i)} - 0.4) \right\rfloor - \left\lfloor (1.6\varepsilon)^{-1} \cdot (x_3^{(j)} - 0.4) \right\rfloor \right| \leq 1.$$

Let  $\hat{\mathbf{x}}^{(i)} = (\lfloor (1.6\varepsilon)^{-1} \cdot (x_2^{(i)} - 0.4) \rfloor, \lfloor (1.6\varepsilon)^{-1} \cdot (x_3^{(i)} - 0.1) \rfloor) \in \mathcal{Q}_{n-3}^2$  for each  $i \in [3]$ . Because [Algorithm 1](#) gives  $C(\mathbf{x}^{(i)}) = C_{\text{rect}}(\hat{\mathbf{x}}^{(i)})$ , we have  $C_{\text{rect}}(\hat{\mathbf{x}}^{(i)}) = i$  for each  $i \in [3]$ . Therefore, we can obtain a solution,  $(\hat{\mathbf{x}}^{(1)}, \hat{\mathbf{x}}^{(2)}, \hat{\mathbf{x}}^{(3)})$ , for the 2-rectSPERNER instance  $C_{\text{rect}}$  in polynomial-time.  $\square$

## 6.2 The coordinate converter

In this subsection, we will define the coordinate converter for this family of instances. From now on, because we have a clear context about the base instance  $C$ , we will assume that  $C$  is any fixed 2D-SPERNER instance constructed by [Algorithm 1](#). Further, we assume that any function we will define has oracle access to this fixed instance  $C$ . Then, the coordinate converter will map any point in the 2-simplex to a point in the 1-simplex, according to  $C$ . This coordinate converter will be used in the next subsection to recursively generate a  $(k + 1)$ -dimensional instance from a  $k$ -dimensional instance.

Before giving the explicit definition, we first give some motivations for why we define this coordinate converter. During our construction, we will repeatedly use the following projection step that maps a point in  $\mathbf{x} \in \Delta^k$  to a point  $\mathbf{P}(\mathbf{x}) \in \Delta^{k-1}$  that is proportional to the first  $k$  coordinates of  $\mathbf{x}$ .

**Definition 29** (Projection step). For any  $k \geq 2$  and any point  $\mathbf{x} \in \Delta^k \setminus \{\mathbf{0}^k, 1\}$ , its projection step  $\mathbf{P} : \Delta^k \rightarrow \Delta^{k-1}$  is defined as  $\mathbf{P}(\mathbf{x}) := \mathbf{x}_{1:k} / (1 - x_{k+1})$ . In particular, we define  $\mathbf{P}(\mathbf{0}^k, 1) = (\mathbf{0}^{k-1}, 1)$ .

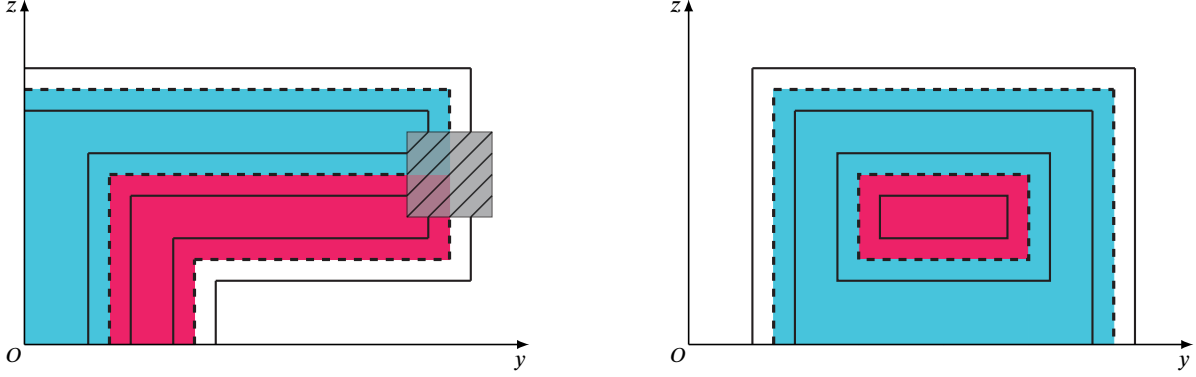


Figure 2: A demonstration of the first steps for our definition of the coordinate converter. Here, we only focus on the region of the coloring that is at most bichromatic because trichromatic regions (e.g., the shadowed part on the left) are PPAD-hard to find. The dashed 1-manifolds represent the “color switches” in each instance. We will find the  $\ell_\infty$  distance (only in terms of the last two coordinates) of each point to these color switches. The solid 1-manifolds represent the sets of points at distance  $\varepsilon^2$  to color switches.

Consider we are now constructing the 3-dimensional instance  $C^{(3)}$ . We can rewrite any point  $\mathbf{x} \in \Delta^3$  in the following form:  $\mathbf{x} = ((1 - x_4) \cdot \mathbf{P}(\mathbf{x}), x_4)$ . A trivial way to define  $C^{(3)}$  is that we use  $C^{(2)}$  at “the bottom” of the 3-simplex, and use the next color uniformly “on top” of it. That is, we let  $C^{(3)}(\mathbf{x}) = C^{(2)}(\mathbf{P}(\mathbf{x}))$  if  $x_4 < 0.1$  and let  $C^{(3)}(\mathbf{x}) = 4$  otherwise. In this trivial definition, it is PPAD-hard to find a small *rainbow* (i.e. *tetrachromatic*) 3-simplex. However, because  $C^{(2)}$  has a lot of regions that are near the “color switches” between two colors and can be found trivially, we have created a lot of spurious easy-to-find trichromatic regions in  $C^{(3)}$ , i.e. at the intersection of  $C^{(2)}$  color switches and the boundary ( $x_4 = 0.1$ ) of the new 4-th color.

In our actual construction, we again start with a  $C^{(2)}$  instance at the “bottom” facet of the 3-simplex. But now, we place another, “standing” copy of the base instance  $C$  on top of each color switch in the bottom copy of  $C^{(2)}$ . Note that, locally, each color switch region in the bottom  $C^{(2)}$  instance is a 1-dimensional manifold. If we look at a short segment perpendicular to this manifold, it’s left half is colored by one color, and it’s right half by the other: exactly like the base of  $C$ ! So given the projection in the bottom  $C^{(2)}$  instance that is close to a color switch, we want to map it to a point  $\tilde{\mathbf{x}} = (1 - \tilde{x}_2, \tilde{x}_2) \in \Delta^1$  at the base of the “top/standing”  $C$  instance. Then, we can combine the “converted” coordinate  $\tilde{x}_2$  and next original coordinate  $x_4$  as  $((1 - x_4) \cdot (1 - \tilde{x}_2), (1 - x_4) \cdot \tilde{x}_2, x_4)$ . Now we can use  $C$ ’s color at  $((1 - x_4) \cdot (1 - \tilde{x}_2), (1 - x_4) \cdot \tilde{x}_2, x_4)$  (properly shifted as will be explained later) to color the corresponding point in  $C^{(3)}$ .

We provide a two-step demonstration of how to define this coordinate converter in Figures 2 and 3. The details of how it is used for  $C^{(3)}$  are provided in Algorithm 2 (next subsection).

**Definition.** Mathematically, the coordinate converter is defined via the nearest differently colored points. More specifically, we consider the  $\ell_\infty$  distance on the second and the third coordinate:

$$d(\mathbf{x}, \mathbf{y}) := \max \{ |x_2 - y_2|, |x_3 - y_3| \} \quad (1)$$

For each point  $\mathbf{x} \in \Delta^2$ , we find the infimum distance among all points with a fixed color  $c \neq C(\mathbf{x})$ :

$$d_{\min}(\mathbf{x}, c) = \inf_{\mathbf{y} \in \Delta^2: C(\mathbf{y})=c} d(\mathbf{x}, \mathbf{y}), \quad (2)$$

and we define its *neighboring color* by

$$C_{\text{nn}}(\mathbf{x}) = \arg \min_{c \in [3]: c \neq C(\mathbf{x})} (d_{\min}(\mathbf{x}, c), c). \quad (3)$$

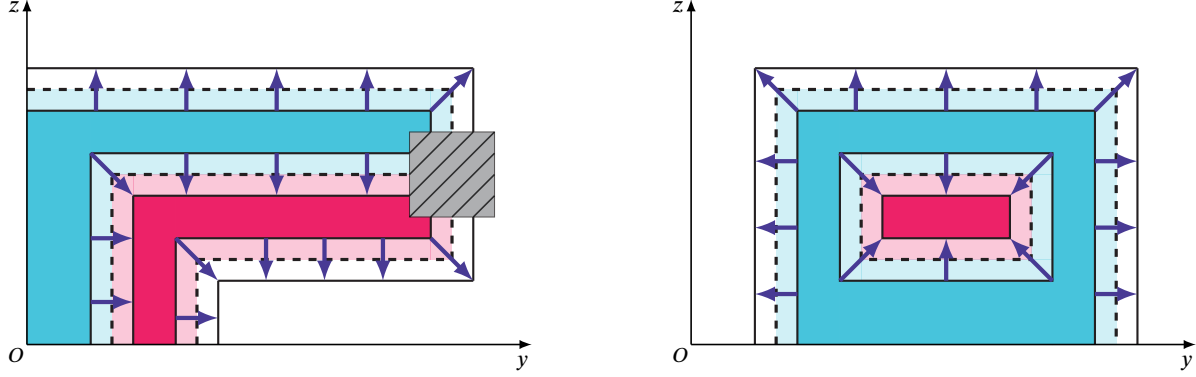


Figure 3: A demonstration of the second steps for our definition of the coordinate converter. Here, we focus on the region of the coloring that is at most bichromatic. Suppose the colors of the blue / red / white regions are respectively indexed by 1 / 2 / 3. For each “color switch”, we draw arrows from the 1-manifold that is  $\epsilon^2$  away from it and has the lower index to the 1-manifold that is  $\epsilon^2$  away from it and has the higher index. The region covered by the arrows are *hot regions*. Points on the sources of the arrows will have a converted coordinate of 0, while those on the sinks will have a converted coordinate of 1. We give converted coordinates for other points on the arrows according to their distances to the sources. Points not on any arrow will be marked as either *warm* or *cold*.

That is, its neighboring color is defined as the color  $c$  that takes the minimum infimum distance  $d_{\min}(\mathbf{x}, c)$ , breaking ties by choosing the color with the smallest index. Or equivalently, the neighboring color of a point equals the different color on the nearest “color switch” from the point. With this definition of neighboring colors, we define the nearest neighbor as the point that attains the minimum in Eq. (2), i.e.:

$$\mathbf{nn}(\mathbf{x}) = \arg \inf_{y \in \Delta^2 : C(y) = C_{\mathbf{nn}}(\mathbf{x}, C)} d(\mathbf{x}, y). \quad (4)$$

Note that the color on the nearest neighbor of  $\mathbf{x}$  may be the same as the color on  $\mathbf{x}$ , but it must correspond to the limit of an infinite sequence of points that have different colors than  $\mathbf{x}$ . Finally, we define the coordinate converter  $\text{rel}^{\mathbf{nn}}$  as follows. (The coordinate converter is defined *relative to the nearest neighbor*, hence its name.) We divide the entire 2-simplex into different regions based on each point’s *distance to the nearest neighbor*, i.e.,  $d(\mathbf{x}, \mathbf{nn}(\mathbf{x}))$ .

- The *hot* region consists of points having distance to the nearest neighbor strictly less than  $\epsilon^2$ , i.e., points  $\mathbf{x}$  with  $d(\mathbf{x}, \mathbf{nn}(\mathbf{x})) < \epsilon^2$ ; we say that points in this region are *hot*.
- The *warm* region consists of points having distance to the nearest neighbor between  $\epsilon^2$  (inclusive) and  $2\epsilon^2$  (exclusive), i.e., points  $\mathbf{x}$  with  $d(\mathbf{x}, \mathbf{nn}(\mathbf{x})) \in [\epsilon^2, 2\epsilon^2)$ ; we say that points in this region are *warm*.
- The *cold* region consists of points having distance to the nearest neighbor no less than  $2\epsilon^2$ , i.e., points  $\mathbf{x}$  with  $d(\mathbf{x}, \mathbf{nn}(\mathbf{x})) \geq 2\epsilon^2$ ; we say that points in this region are *cold*.

For points in the *hot* region induced by each “color switch”, we map it continuously to a converted coordinate in  $(0, 1)$ , from the boundary with a smaller-indexed color to the boundary with a larger-indexed color. For points in the *warm* region induced by each “color switch”, we integrally map it to 0 or 1 depending on whether its color has a smaller or larger index than that of its neighboring color. For points in the *cold* region, we enforce its converted coordinate to 0 or 1 only depending on its own color. We summarize the definition in Eq. (5), where the first two cases are for the *hot* and *warm* regions and the last two cases are for the *cold*

region.

$$\text{rel}^{\text{nn}}(\mathbf{x}) = \begin{cases} (0.5 - 0.5\epsilon^{-2} \cdot d(\mathbf{x}, \text{nn}(\mathbf{x})))_+ & \text{if } \mathbf{x} \text{ is hot/warm and } C_{\text{nn}}(\mathbf{x}) > C(\mathbf{x}), \\ (0.5 + 0.5\epsilon^{-2} \cdot d(\mathbf{x}, \text{nn}(\mathbf{x})))_- & \text{if } \mathbf{x} \text{ is hot/warm and } C_{\text{nn}}(\mathbf{x}) < C(\mathbf{x}), \\ 0 & \text{if } \mathbf{x} \text{ is cold and } C(\mathbf{x}) = 1, \\ 1 & \text{if } \mathbf{x} \text{ is cold and } C(\mathbf{x}) \in \{2, 3\}. \end{cases} \quad (5)$$

Because warm points satisfy  $d(\mathbf{x}, \text{nn}(\mathbf{x})) \geq \epsilon^2$  and hot points are those with  $d(\mathbf{x}, \text{nn}(\mathbf{x})) < \epsilon^2$ , we have the following fact.

**Fact 30.** *For any point  $\mathbf{x} \in \Delta^2$ , we have  $\text{rel}^{\text{nn}}(\mathbf{x}) \in (0, 1)$  if and only if  $\mathbf{x}$  is hot.*

As discussed above, our use of this coordinate converter is to convert each point  $\mathbf{x} \in \Delta^3$  to  $((1 - x_4) \cdot (1 - \text{rel}^{\text{nn}}(\mathbf{P}(\mathbf{x}))), (1 - x_4) \cdot \text{rel}^{\text{nn}}(\mathbf{P}(\mathbf{x})), x_4) \in \Delta^2$ . If  $\mathbf{P}(\mathbf{x})$  is warm or cold,  $\text{rel}^{\text{nn}}(\mathbf{P}(\mathbf{x})) = 0$  and  $\mathbf{x}$  will be converted to either  $(1 - z, 0, z)$  or  $(0, 1 - z, z)$  for  $z = x_4$ . Recall [Fact 26](#), where we have shown for  $C$  that

$$\forall z \in [0, 1], \quad C(1 - z, 0, z) = \begin{cases} 1 & z \leq 0.1, \\ 3 & z > 0.1; \end{cases} \quad C(0, 1 - z, z) = \begin{cases} 2 & z \leq 0.1, \\ 3 & z > 0.1. \end{cases}$$

One unified way to interpret this coloring is to consider the colors on the line segments  $(1 - z, 0, z)$  and  $(0, 1 - z, z)$  as functions of  $z \in [0, 1]$ : when  $z > 0.1$  they take their color from  $(0, 0, 1)$ , and when  $z \leq 0.1$  they take the color of the other endpoint. This symmetry will be useful later when we show that there are no spurious solutions created where two warm/cold points are close to each other but have a different converted coordinate (i.e., one equals 0, but the other equals 1). Therefore, we can consider the following equivalence between the converted coordinates.

**Definition 31** (Equivalence between converted coordinates). *For any converted coordinates  $\tilde{x}, \tilde{x}' \in [0, 1]$ , we say they are equivalent (denoted as  $\tilde{x} \sim_{\text{rel}} \tilde{x}'$ ) if we have either  $\tilde{x} = \tilde{x}'$  or  $\tilde{x}, \tilde{x}' \in \{0, 1\}$ .*

**Claim 32.** *For any converted coordinates  $\tilde{x}, \tilde{x}' \in [0, 1]$  such that  $\tilde{x} \sim \tilde{x}'$ , and any  $z \in [0, 1]$ , we have*

$$C((1 - z) \cdot (1 - \tilde{x}), (1 - z) \cdot \tilde{x}, z) = C((1 - z) \cdot (1 - \tilde{x}'), (1 - z) \cdot \tilde{x}', z).$$

## 6.2.1 Key properties of the coordinate converter

Next, we establish the key properties about the coordinate converter on this family of instances.

**Property I: Polynomial time computation** First, we show that all previous functions can be “roughly” implemented in polynomial time and we can thus use them freely in our reduction. That is, we can always output the true converted coordinate  $\text{rel}^{\text{nn}}$  and, whenever  $\mathbf{x}$  is hot or warm (i.e.,  $d(\mathbf{x}, \text{nn}(\mathbf{x})) < 2\epsilon^2$ ), we can also output the neighboring color  $C_{\text{nn}}(\mathbf{x})$ .

**Lemma 33.** *Given oracle access to  $C$ , there is a polynomial-time algorithm that takes any  $\mathbf{x} \in \Delta^2$  as input and that outputs  $\text{rel}^{\text{nn}}(\mathbf{x})$ . Furthermore, if  $d(\mathbf{x}, \text{nn}(\mathbf{x})) < 2\epsilon^2$ , the algorithm can also compute  $C_{\text{nn}}(\mathbf{x})$  in polynomial time.*

*Proof.* According to our definition of the coordinate converter [Eq. \(5\)](#), it suffices to only compute the exact value of  $d(\mathbf{x}, \text{nn}(\mathbf{x}))$  for  $\mathbf{x}$  such that  $d(\mathbf{x}, \text{nn}(\mathbf{x})) \leq 2\epsilon^2$ . The value of  $d(\mathbf{x}, \text{nn}(\mathbf{x}))$  can be computed by its distance to each “color switch”. Note that we only have a constant number of all the color switches outside the core region are defined by  $O(1)$  line segments ([Algorithm 1](#), or see [Figure 1](#)), thus we can enumerate over them. For those “color switches” strictly inside the core region, because each point in  $C_{\text{rect}}$  is transformed into a  $1.6\epsilon \times 1.6\epsilon$  square ([Algorithm 1](#)) and  $0.8\epsilon \gg 2\epsilon^2$ , we only need to check at most 8 points in the 2D-rectSPERNER instance  $C_{\text{rect}}$ , which can be done by querying the oracle of  $C$  according to [Algorithm 1](#).  $\square$



**Property II: Symmetry** As mentioned earlier, we will use the converted coordinate with a new coordinate (in a higher dimension) to embed a new copy of the 2D-SPERNER instance inside our construction. That is, supposing the converted coordinate is  $\tilde{x}$  and the new coordinate is  $z$ , we will define a vector,  $((1 - z) \cdot (1 - \tilde{x}), (1 - z) \cdot \tilde{x}, z)$ , for the new copy. Later, to argue that there is no issue arising from the fact that we can have very close points with a converted coordinate of  $\tilde{x} = 0$  and of  $\tilde{x} = 1$ , we also need the following property that the converted coordinates of  $(1 - z, 0, z)$  and  $(0, 1 - z, z)$  are equivalent and their neighboring colors are symmetric. This stronger symmetry also implies a characterization of the temperature on the boundary:

- If  $z \in 0.1 \pm \varepsilon^2$ , points  $(1 - z, 0, z)$  and  $(0, 1 - z, z)$  are *hot* and the converted coordinate equals  $0.5 + 0.5\varepsilon^{-2} \cdot (z - 0.1)$ .
- If  $z \in 0.1 \pm 2\varepsilon^2$  but  $z \notin 0.1 \pm \varepsilon^2$ , points  $(1 - z, 0, z)$  and  $(0, 1 - z, z)$  are *warm* and the converted coordinate is the rounding of  $0.5 + 0.5\varepsilon^{-2} \cdot (z - 0.1)$  to its nearest integer in  $\{0, 1\}$ .
- If  $z \notin 0.1 \pm 2\varepsilon^2$ , points  $(1 - z, 0, z)$  and  $(0, 1 - z, z)$  are *cold* and the converted coordinate is in  $\{0, 1\}$  (where we don't care its exact value).

Here, the value of  $|z - 0.1|$  gives the *hot* and *warm* points' distances to their nearest neighbor.

**Lemma 34** (Symmetry on converted coordinate). *Warm and hot points on the line segments  $(1 - x_3, 0, x_3)$  and  $(0, 1 - x_3, x_3)$  have  $x_3 \in 0.1 \pm 2\varepsilon^2$ . For any such  $x_3$ , we have*

$$\text{rel}^{\text{nn}}(1 - x_3, 0, x_3) = \left( 1/2 + \frac{x_3 - 0.1}{2\varepsilon^2} \right)_{[0,1]} = \text{rel}^{\text{nn}}(0, 1 - x_3, x_3), \quad (6)$$

and the neighboring color is characterized as follows

$$C_{\text{nn}}(1 - x_3, 0, x_3) = \begin{cases} 3 & \text{if } x_3 \leq 0.1, \\ 1 & \text{if } x_3 > 0.1, \end{cases} \quad \text{and} \quad C_{\text{nn}}(0, 1 - x_3, x_3) = \begin{cases} 3 & \text{if } x_3 \leq 0.1, \\ 2 & \text{if } x_3 > 0.1. \end{cases}$$

*Proof.* Note that we guarantee our coloring to satisfy all the following equations.

$$\forall \mathbf{y} \in \Delta^2, \quad \begin{cases} C(\mathbf{y}) = 1 & \text{if } y_3 \leq 0.1, y_2 \leq 0.5, \\ C(\mathbf{y}) = 2 & \text{if } y_3 \leq 0.1, y_2 > 0.5, \\ C(\mathbf{y}) = 3 & \text{if } y_3 > 0.1, y_2 \notin 0.5 \pm 0.1, \\ C(\mathbf{y}) = 3 & \text{if } y_3 > 0.3. \end{cases}$$

**Consider any  $x_3 \leq 0.1$ .** We have  $C(1 - x_3, 0, x_3) = 1$  and  $C(0, 1 - x_3, x_3) = 2$ . Note that any point  $\mathbf{y}$  that has color other than 1 has either  $y_3 \geq 0.1$  or  $y_2 > 0.5$ . Hence, we have  $\text{nn}(1 - x_3, 0, x_3) = (0.9, 0, 0.1)$  for any  $x_3 \leq 0.1$ . We then have  $d((1 - x_3, 0, x_3), \text{nn}(1 - x_3, 0, x_3)) = 0.1 - x_3$  and  $C_{\text{nn}}(1 - x_3, 0, x_3) = 3$ . According to the definition of the coordinate converter Eq. (5), if  $0.1 - x_3 < 2\varepsilon^2$  (i.e.,  $x_3 > 0.1 - 2\varepsilon^2$ ), the point  $(1 - x_3, 0, x_3)$  is hot or warm, and

$$\text{rel}^{\text{nn}}(1 - x_3, 0, x_3) = (0.5 - 0.5\varepsilon^{-2} \cdot (0.1 - x_3))_+ = \left( 0.5 + \frac{x_3 - 0.1}{2\varepsilon^2} \right)_{[0,1]},$$

which matches Eq. (6). Otherwise, we have  $d((1 - x_3, 0, x_3), \text{nn}(1 - x_3, 0, x_3)) \geq 2\varepsilon^2$  and thus the point  $(1 - x_3, 0, x_3)$  is cold. Similarly, for any  $x_3 \leq 0.1$ , we have  $\text{nn}(0, 1 - x_3, x_3) = (0, 0.9, 0.1)$ ,  $d((0, 1 - x_3, x_3), \text{nn}(0, 1 - x_3, x_3)) = 0.1 - x_3$ , and  $C_{\text{nn}}(0, 1 - x_3, x_3) = 3$ . According to the definition of the temperature, if  $x_3 > 0.1 - 2\varepsilon^2$ , the point  $(0, 1 - x_3, x_3)$  is hot or warm, and we can obtain the right-handed-side equation of Eq. (6) according to Eq. (5). Otherwise, we have  $d((0, 1 - x_3, x_3), \text{nn}(0, 1 - x_3, x_3)) \geq 2\varepsilon^2$ , which implies that the point  $(0, 1 - x_3, x_3)$  is cold.

**Consider any**  $x_3 > 0.1$ . We have  $C(1 - x_3, 0, x_3) = C(0, 1 - x_3, x_3) = 3$ . Note that any point  $\mathbf{y}$  that has color other than 3 has either  $y_3 \leq 0.1$  or  $y_2 \in [0.4, 0.6]$ . Hence, if  $x_3 \geq 0.1 + 2\epsilon^2$ , we have  $d((1 - x_3, 0, x_3), \mathbf{nn}(1 - x_3, 0, x_3)) \geq 2\epsilon^2$  and thus the point  $(1 - x_3, 0, x_3)$  is cold. Otherwise, if  $x_3 < 0.1 + 2\epsilon^2$ , we have  $\mathbf{nn}(1 - x_3, 0, x_3) = (0.9, 0, 0.1)$ . We then have  $d((1 - x_3, 0, x_3), \mathbf{nn}(1 - x_3, 0, x_3)) = z - 0.1 < 2\epsilon^2$  and  $C_{\mathbf{nn}}(1 - x_3, 0, x_3) = 1$ . The point  $(1 - x_3, 0, x_3)$  is hot or warm in this case. Further, according to the definition of the coordinate converter [Eq. \(5\)](#),

$$\text{rel}^{\mathbf{nn}}(1 - x_3, 0, x_3) = (0.5 + 0.5\epsilon^{-2} \cdot (x_3 - 0.1))_{-} = \left(0.5 + \frac{x_3 - 0.1}{2\epsilon^2}\right)_{[0,1]},$$

matching [Eq. \(6\)](#). On the other hand, note that any point  $\mathbf{y}$  has color other than 3 and  $y_3 \geq 0.1$  must have  $y_2 + y_3 \leq 0.9$ . The distance  $d$  between  $(0, 1 - x_3, x_3)$  and any such point is at least 0.05. Note that  $0.05 \gg 2\epsilon^2$ . Hence, when  $x_3 \geq 0.1 + 2\epsilon^2$ , we have  $d((0, 1 - x_3, x_3), \mathbf{nn}(0, 1 - x_3, x_3)) \geq 2\epsilon^2$  and thus according to the definition of the temperature, the point  $(0, 1 - x_3, x_3)$  is cold. Otherwise, if  $x_3 < 0.1 + 2\epsilon^2$ , we have  $\mathbf{nn}(0, 1 - x_3, x_3) = (x_3 - 0.1, 1 - x_3, 0.1)$ . We then have  $d((0, 1 - x_3, x_3), \mathbf{nn}(0, 1 - x_3, x_3)) = x_3 - 0.1$  and  $C_{\mathbf{nn}}(0, 1 - x_3, x_3) = 2$ . According to the definition of the coordinate converter [Eq. \(5\)](#),

$$\text{rel}^{\mathbf{nn}}(0, 1 - x_3, x_3) = (0.5 + 0.5\epsilon^{-2} \cdot (x_3 - 0.1))_{-} = \left(0.5 + \frac{x_3 - 0.1}{2\epsilon^2}\right)_{[0,1]},$$

matching [Eq. \(6\)](#). □

**Corollary 35.** *For any converted coordinates  $\tilde{x}, \tilde{x}' \in [0, 1]$  such that  $\tilde{x} \sim \tilde{x}'$ , and any  $z \in [0, 1]$ , we have*

$$\text{rel}^{\mathbf{nn}}((1 - z) \cdot (1 - \tilde{x}), (1 - z) \cdot \tilde{x}, z) \sim_{\text{rel}} \text{rel}^{\mathbf{nn}}((1 - z) \cdot (1 - \tilde{x}'), (1 - z) \cdot \tilde{x}', z).$$

**Property III: Lipschitzness** We show that the coordinate converter is Lipschitz in bichromatic regions in the following sense: consider we view the interval  $[0, 1]$  as a loop, where 0, 1 corresponds to the same point on the loop, and the distance between any  $\text{rel}^{\mathbf{nn}}(\mathbf{x})$  and  $\text{rel}^{\mathbf{nn}}(\mathbf{x}')$  on the loop can be upper bounded by a  $\text{poly}(1/\epsilon)$  factor of the distance between  $\mathbf{x}$  and  $\mathbf{x}'$ . We say a point is in a bichromatic region if its neighbourhood, defined in [Definition 36](#), is bichromatic. And, the Lipschitzness property is formalized by [Lemma 37](#) with a slightly stronger argument.

**Definition 36.** *Let  $\epsilon = 2^{-n}$ . For any  $\mathbf{x} \in \Delta^2$ , let  $\mathcal{N}(\mathbf{x}) = \{\mathbf{y} \in \Delta^2 : |x_2 - y_2|, |x_3 - y_3| \leq \epsilon/2\}$  denote the neighbourhood of  $\mathbf{x}$  with a side length of  $\epsilon$ .*

**Lemma 37.** *For any coloring  $C$  and any pair of points  $\mathbf{x}, \mathbf{x}'$ , at least one of the following properties is satisfied:*

- **one is in a trichromatic region:** *there are three different colors in  $\mathcal{N}(\mathbf{x})$  (or,  $\mathcal{N}(\mathbf{x}')$ );*
- **Lipschitz in the hot&warm regions:**  $|\text{rel}^{\mathbf{nn}}(\mathbf{x}) - \text{rel}^{\mathbf{nn}}(\mathbf{x}')| \leq \epsilon^{-2} \cdot \|\mathbf{x} - \mathbf{x}'\|_{\infty}$ ;
- **both are in the warm&cold regions:**  $\text{rel}^{\mathbf{nn}}(\mathbf{x}), \text{rel}^{\mathbf{nn}}(\mathbf{x}') \in \{0, 1\}$ .

*Proof.* It suffices to show that the Lipschitz property,  $|\text{rel}^{\mathbf{nn}}(\mathbf{x}) - \text{rel}^{\mathbf{nn}}(\mathbf{x}')| \leq \epsilon^{-2} \cdot \|\mathbf{x} - \mathbf{x}'\|_{\infty}$ , under the following three assumptions: (1)  $\mathcal{N}(\mathbf{x})$  and  $\mathcal{N}(\mathbf{x}')$  are both at most bichromatic; (2)  $\|\mathbf{x} - \mathbf{x}'\|_{\infty} < \epsilon^2$ ; and (3)  $\text{rel}^{\mathbf{nn}}(\mathbf{x}) \in (0, 1)$  (equivalently by [Fact 30](#),  $d(\mathbf{x}, \mathbf{nn}(\mathbf{x})) < \epsilon^2$ ). The second assumption makes sense because  $\text{rel}^{\mathbf{nn}}(\mathbf{x}), \text{rel}^{\mathbf{nn}}(\mathbf{x}') \in [0, 1]$  and the second property can trivially hold if the assumption is violated. In addition, the second and the third assumption ensures that  $d(\mathbf{x}, \mathbf{nn}(\mathbf{x})), d(\mathbf{x}', \mathbf{nn}(\mathbf{x}')) < 2\epsilon^2$ , which can be obtained by the triangle inequality when  $\mathbf{x}, \mathbf{x}'$  have the same color and otherwise is trivial by the second assumption. This ensures that we must go to the first two cases of [Eq. \(5\)](#) when computing the converted coordinates of  $\mathbf{x}, \mathbf{x}'$ . Next, we discuss two cases on the colors of  $\mathbf{x}$  and  $\mathbf{x}'$ .

**Case 1:  $\mathbf{x}$  and  $\mathbf{x}'$  have different colors.** Since  $d(\mathbf{x}, \mathbf{x}') \leq \|\mathbf{x} - \mathbf{x}'\|_\infty$ , we have  $d(\mathbf{x}, \mathbf{nn}(\mathbf{x})) \leq \|\mathbf{x} - \mathbf{x}'\|_\infty$  and  $d(\mathbf{x}', \mathbf{nn}(\mathbf{x}')) \leq \|\mathbf{x} - \mathbf{x}'\|_\infty$ . According to [Eq. \(5\)](#), we have

$$\begin{aligned} \text{rel}^{\text{nn}}(\mathbf{x}) &\in 0.5 \pm 0.5\varepsilon^{-2} \cdot d(\mathbf{x}, \mathbf{nn}(\mathbf{x})) \subseteq 0.5 \pm 0.5\varepsilon^{-2} \cdot \|\mathbf{x} - \mathbf{x}'\|_\infty, \quad \text{and} \\ \text{rel}^{\text{nn}}(\mathbf{x}') &\in 0.5 \pm 0.5\varepsilon^{-2} \cdot d(\mathbf{x}', \mathbf{nn}(\mathbf{x}')) \subseteq 0.5 \pm 0.5\varepsilon^{-2} \cdot \|\mathbf{x} - \mathbf{x}'\|_\infty. \end{aligned}$$

Therefore, we have  $|\text{rel}^{\text{nn}}(\mathbf{x}) - \text{rel}^{\text{nn}}(\mathbf{x}')| \leq \varepsilon^{-2} \cdot \|\mathbf{x} - \mathbf{x}'\|_\infty$ .

**Case 2:  $\mathbf{x}$  and  $\mathbf{x}'$  have the same color.** Suppose that  $c := C(\mathbf{x}) = C(\mathbf{x}')$ . Because of the definition of the coordinate converter ([Eq. \(5\)](#)), the third assumption implies  $d(\mathbf{x}, \mathbf{nn}(\mathbf{x})) < \varepsilon^2$ . Since  $C(\mathbf{x}) \neq C_{\text{nn}}(\mathbf{x})$ , according to the definition of the neighbourhood  $\mathcal{N}(\mathbf{x})$  ([Definition 36](#)), this further implies that  $\mathcal{N}(\mathbf{x})$  is bichromatic combined with our first assumption. Suppose that  $c'$  is the other color in  $\mathcal{N}(\mathbf{x})$ . Or say,  $c' := C_{\text{nn}}(\mathbf{x})$ . Recall that we have showed that  $d(\mathbf{x}', \mathbf{nn}(\mathbf{x}')) < 2\varepsilon^2$ . Note that the following set is strictly contained in  $\mathcal{N}(\mathbf{x})$ :

$$\{\mathbf{y} \in \Delta^2 : y_2 \in x'_2 \pm 2\varepsilon^2, y_3 \in x'_3 \pm 2\varepsilon^2\}.$$

We don't have any color other than  $c, c'$  within a distance of  $\varepsilon^2$  from  $\mathbf{x}'$ , and thus  $C_{\text{nn}}(\mathbf{x}') = c'$ . This means

$$\begin{aligned} d(\mathbf{x}, \mathbf{nn}(\mathbf{x})) &= d_{\min}(\mathbf{x}, c') = \inf_{\mathbf{y} \in \Delta^2 : C(\mathbf{y})=c'} d(\mathbf{x}, \mathbf{y}), \\ d(\mathbf{x}', \mathbf{nn}(\mathbf{x}')) &= d_{\min}(\mathbf{x}', c') = \inf_{\mathbf{y} \in \Delta^2 : C(\mathbf{y})=c'} d(\mathbf{x}', \mathbf{y}). \end{aligned}$$

Therefore, we have

$$\begin{aligned} |d(\mathbf{x}, \mathbf{nn}(\mathbf{x})) - d(\mathbf{x}', \mathbf{nn}(\mathbf{x}'))| &= \left| \inf_{\mathbf{y} \in \Delta^2 : C(\mathbf{y})=c'} d(\mathbf{x}, \mathbf{y}) - \inf_{\mathbf{y} \in \Delta^2 : C(\mathbf{y})=c'} d(\mathbf{x}', \mathbf{y}) \right| \\ &\leq \inf_{\mathbf{y} \in \Delta^2 : C(\mathbf{y})=c'} |d(\mathbf{x}, \mathbf{y}) - d(\mathbf{x}', \mathbf{y})| \\ &\leq 2 \|\mathbf{x} - \mathbf{x}'\|_\infty. \end{aligned}$$

Because  $C(\mathbf{x}) = c = C(\mathbf{x}')$  and  $C_{\text{nn}}(\mathbf{x}) = c' = C_{\text{nn}}(\mathbf{x}')$ , when we compute  $\text{rel}^{\text{nn}}(\mathbf{x})$  and  $\text{rel}^{\text{nn}}(\mathbf{x}')$ , we follow the same case of [Eq. \(5\)](#). Hence,

$$|\text{rel}^{\text{nn}}(\mathbf{x}) - \text{rel}^{\text{nn}}(\mathbf{x}')| \leq 0.5\varepsilon^{-2} \cdot |d(\mathbf{x}, \mathbf{nn}(\mathbf{x})) - d(\mathbf{x}', \mathbf{nn}(\mathbf{x}'))| \leq \varepsilon^{-2} \cdot \|\mathbf{x} - \mathbf{x}'\|_\infty,$$

and we have obtained the lemma for this case.  $\square$

Furthermore, a very similar (and slightly simpler) Lipschitzness can be established for points on the two boundaries other than the base 1-simplex. The proof is much simpler – we only need to do simple calculations based on our previous characterization of the coordinate converter ([Lemma 34](#)). The technical proof is deferred to [Appendix A.2](#).

**Lemma 38.** *For any  $z, z' \in [0, 1]$  and any pair of points  $\mathbf{x}, \mathbf{x}'$  such that  $\mathbf{x} \in \{(0, 1 - z, z), (1 - z, 0, z)\}$  and  $\mathbf{x}' \in \{(0, 1 - z', z'), (1 - z', 0, z')\}$ , at least one of the following properties is satisfied:*

- **Lipschitz in the hot&warm regions:**  $|\text{rel}^{\text{nn}}(\mathbf{x}) - \text{rel}^{\text{nn}}(\mathbf{x}')| \leq \varepsilon^{-2} \cdot |z - z'|$ ;
- **both are in the warm&cold regions:**  $\text{rel}^{\text{nn}}(\mathbf{x}), \text{rel}^{\text{nn}}(\mathbf{x}') \in \{0, 1\}$ .

**Property IV: Cold on the top** Finally, we show that the points that have reasonably high values on the third dimension ( $x_3 > 0.5$ ) are all cold. This lemma will help ensure that there are no spurious solutions in which any of the three points has a coordinate that has a very high value (e.g.,  $> 1 - 2^{-\Omega(n)}$ ) in any of its  $\ell$ -th projection step (see [Definition 40](#) and next subsection for details).

**Fact 39.** For any  $\mathbf{x} \in \Delta^2$  such that  $x_3 \geq 0.5$ , we have  $\text{rel}^{nn}(\mathbf{x}) = 1$ .

*Proof.* According to [Algorithm 1](#), we have  $C(\mathbf{x}) = 3$  for any  $\mathbf{x} \in \Delta^2$  such that  $x_3 > 0.3$ . Therefore, if  $x_3 \geq 0.5$ ,  $d(\mathbf{x}, nn(\mathbf{x})) \geq 0.2 \gg 2\epsilon^2$  and thus  $\mathbf{x}$  is cold. According to [Eq. \(5\)](#),  $\text{rel}^{nn}(\mathbf{x}) = 1$ .  $\square$

### 6.3 Hard instances with three or more dimensions

In this subsection, we give the constructions for  $C^{(k)}$  with  $k \geq 3$ . Before giving the construction, for convenience, we introduce the  $\ell$ -th projection steps we will recursively use to project a point.

**Definition 40** ( $\ell$ -th projection steps). For any  $0 \leq \ell \leq k - 1$ , we use  $\mathbf{P}^{(\ell)}(\mathbf{x}) \in \Delta^{k-\ell}$  to denote the vector we obtain after applying a total number of  $\ell$  projection steps on  $\mathbf{x}$ , i.e., we let  $\mathbf{P}^{(0)}(\mathbf{x}) = \mathbf{x}$  and let  $\mathbf{P}^{(\ell)}(\mathbf{x}) = \mathbf{P}(\mathbf{P}^{(\ell-1)}(\mathbf{x}))$  for any  $\ell \geq 1$ .

We also define a *modified neighbor color* that is consistent with our definition of the coordinate converter for those cold points  $\mathbf{x}$  having  $d(\mathbf{x}, nn(\mathbf{x})) \geq 2\epsilon^2$ .

$$\hat{C}_{nn}(\mathbf{x}) = \begin{cases} C_{nn}(\mathbf{x}) & \text{if } d(\mathbf{x}, nn(\mathbf{x})) < 2\epsilon^2, \\ 2 & \text{if } d(\mathbf{x}, nn(\mathbf{x})) \geq 2\epsilon^2 \text{ and } C(\mathbf{x}) = 1, \\ 1 & \text{if } d(\mathbf{x}, nn(\mathbf{x})) \geq 2\epsilon^2 \text{ and } C(\mathbf{x}) \in \{2, 3\}. \end{cases} \quad (7)$$

The efficient computation of this function follows from [Lemma 33](#). This will be an auxiliary function for our algorithm, which allows us to use the following fact.

**Fact 41.** For any  $\mathbf{x} \in \Delta^2$ , we have  $C(\mathbf{x}) < \hat{C}_{nn}(\mathbf{x})$  if  $\text{rel}^{nn}(\mathbf{x}) = 0$ , and  $C(\mathbf{x}) > \hat{C}_{nn}(\mathbf{x})$  if  $\text{rel}^{nn}(\mathbf{x}) = 1$ .

Our construction (formalized in [Algorithm 2](#)) for three or more dimensions is recursive, i.e., we lift  $C^{(k)}$  from  $C^{(k-1)}$  by the base instance  $C$ . During this process, we recursively compute a sequence of points  $\mathbf{y}^{(2)}, \dots, \mathbf{y}^{(k)} \in \Delta^2$ , that can each be thought of as a different projection of  $\mathbf{x}$  to a 2-simplex; additionally, we compute for each  $i \in \{2, \dots, k\}$  a palette of three candidate colors  $\mathbf{c}^{(i)} \in \binom{[i+1]}{3}$  that can be used to color the  $i$ -th copy of  $C$ ; in particular, we can use  $c_j^{(k)}$  for  $j = C(\mathbf{y}^{(k)})$  as our output color.

In more detail, our construction begins with projecting  $\mathbf{x}$  to the base 2-simplex (i.e., letting  $\mathbf{y}^{(2)} = \mathbf{P}^{(k-2)}(\mathbf{x})$ ) and a vector of the only three colors  $\mathbf{c}^{(2)} = (1, 2, 3)$ . This initialization is consistent with our 2-dimensional instance  $C^{(2)}$  and the base instance  $C$  constructed in [Subsection 6.1](#). Then, we recursively lift the  $(i-1)$ -dimensional instance to the  $i$ -dimensional instance for each  $i \in \{3, \dots, k\}$  by the following steps:

1. We locally project the previous projection  $\mathbf{y}^{(i-1)}$  to a point  $(1 - \tilde{y}, \tilde{y})$  on 1-simplex, using our converted coordinate  $\tilde{y} = \text{rel}^{nn}(\mathbf{y}^{(i-1)})$ .
2. We let the new coordinate  $z$  (i.e., the next coordinate in  $C^{(i)}$ ) be the last coordinate of  $\mathbf{x}$ 's projection to the base  $i$ -simplex (i.e.,  $\mathbf{P}^{(k-i)}(\mathbf{x})$ ). Note that this coordinate  $z$  was not used in the construction of  $C^{(i-1)}$ .
3. With  $\tilde{y}$  and  $z$ , we define our new projection of  $\mathbf{x}$  to a 2-simplex by  $\mathbf{y}^{(i)} = ((1 - z) \cdot (1 - \tilde{y}), (1 - z) \cdot \tilde{y}, z)$ , which can be viewed as an weighted average of our previous converted point  $(1 - \tilde{y}, \tilde{y}, 0)$  on the base 1-simplex and topmost point  $(0, 0, 1)$  via the new coordinate  $z$ .

---

**Algorithm 2:** 3-out-of- $k+1$  Approximate SPERNER Instance  $C^{(k)}(\mathbf{x})$ 


---

**Input** : vector  $\mathbf{x} \in \Delta^k$   
**Output** : color  $c \in [k+1]$

- 1  $\mathbf{y}^{(2)} \leftarrow \mathbf{P}^{(k-2)}(\mathbf{x})$  // initiate  $\mathbf{y}$  by  $\mathbf{x}$ 's projection to the base 2-simplex
- 2  $\mathbf{c}^{(2)} \leftarrow (1, 2, 3)$  // initiate the set of 3 colors
- 3 **for**  $i \in \{3, \dots, k\}$  **do**
- 4  $y_3^{(i)} \leftarrow P_{i+1}^{(k-i)}(\mathbf{x})$  // find the new z-coordinate from the projection of  $\mathbf{x}$
- 5  $y_2^{(i)} \leftarrow (1 - y_3^{(i)}) \cdot \text{rel}^{\text{nn}}(\mathbf{y}^{(i-1)})$
- 6  $y_1^{(i)} \leftarrow (1 - y_3^{(i)}) \cdot (1 - \text{rel}^{\text{nn}}(\mathbf{y}^{(i-1)}))$  // convert  $\mathbf{y}$  to a point on the 2-simplex
- 7  $j_1 \leftarrow \min \{C(\mathbf{y}^{(i-1)}), \hat{C}_{\text{nn}}(\mathbf{y}^{(i-1)})\}$
- 8  $j_2 \leftarrow \max \{C(\mathbf{y}^{(i-1)}), \hat{C}_{\text{nn}}(\mathbf{y}^{(i-1)})\}$
- 9  $\mathbf{c}^{(i)} \leftarrow (c_{j_1}^{(i-1)}, c_{j_2}^{(i-1)}, i+1)$  // obtain the new set of 3 colors
- 10  $j \leftarrow C(\mathbf{y}^{(k)})$
- 11 **return**  $c_j^{(k)}$

---

4. To define the  $i$ -th palette of three new candidate colors, we look at the previous color  $c_j^{(i-1)}$  for  $j = C(\mathbf{y}^{(i-1)})$  and the previous neighboring color  $c_{j'}^{(i-1)}$  for  $j' = C_{\text{nn}}(\mathbf{y}^{(i-1)})$ , and use  $c_j^{(i-1)}, c_{j'}^{(i-1)}$  and the new color  $i+1$  as the three candidate colors.

Because [Algorithm 2](#) is clearly polynomial-time, in the rest of this subsection, we will prove that we can recover a solution of the 2D-SPERNER instance  $C$  in polynomial time from any solution of our instances  $C^{(k)}$  to finish the analysis.

### 6.3.1 Recovering a 2D solution

From now on, we assume that the tuple  $(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \mathbf{x}^{(3)})$  is any good solution for the 3-out-of- $k+1$  Approximate SPERNER instance  $C^{(k)}$  satisfying that  $\|\mathbf{x}^{(i)} - \mathbf{x}^{(j)}\|_\infty \leq 2^{-3kn} = \epsilon^3$  for any  $i, j \in [3]$ . We will prove that we can recover a 2D-SPERNER solution from it in polynomial time.

**Lemma 42.** *Given oracle access to a 2D-SPERNER instance  $C$  and a tuple of points  $(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \mathbf{x}^{(3)})$  which satisfy  $\|\mathbf{x}^{(i)} - \mathbf{x}^{(j)}\|_\infty \leq 2^{-3kn}$  for any  $i, j \in [3]$ , and which are trichromatic in the 3-out-of- $k+1$  Approximate SPERNER instance  $C^{(k)}$  constructed by [Algorithm 2](#), then there is a polynomial-time algorithm that finds a tuple of points  $(\hat{\mathbf{x}}^{(1)}, \hat{\mathbf{x}}^{(2)}, \hat{\mathbf{x}}^{(3)})$  which satisfy  $\|\hat{\mathbf{x}}^{(i)} - \hat{\mathbf{x}}^{(j)}\|_\infty \leq 2^{-n}$  for any  $i, j \in [3]$  and which are trichromatic in the 2D-SPERNER instance  $C$ .*

In the analysis, we use  $\mathbf{y}^{(i)}(\mathbf{x}) = (y_1^{(i)}(\mathbf{x}), y_2^{(i)}(\mathbf{x}), y_3^{(i)}(\mathbf{x}))$  to denote the *intermediate projections* we use in [Algorithm 2](#) when the inputs are  $\mathbf{x}$  and  $C$ . Similarly, we use  $\mathbf{c}^{(i)}(\mathbf{x}) = (c_1^{(i)}(\mathbf{x}), c_2^{(i)}(\mathbf{x}), c_3^{(i)}(\mathbf{x}))$  to denote the *intermediate palettes* we use in [Algorithm 2](#) for  $\mathbf{x}$ . In the rest of this section, because the subscripts we will use for  $\mathbf{c}^{(i)}$  can be very complicated, we will use  $c_j^{(i)}(\mathbf{x})$  and  $c^{(i)}(\mathbf{x}, j)$  interchangeably for better presentation. For any vector  $\mathbf{y} \in \Delta^2$ , we use  $i^*(\mathbf{y})$  to denote the first non-zero index of  $\mathbf{y}$ , i.e.,

$$i^*(\mathbf{y}) = \begin{cases} 1 & \text{if } y_1 > 0, \\ 2 & \text{if } y_1 = 0 \text{ and } y_2 > 0, \\ 3 & \text{otherwise.} \end{cases} \quad (8)$$

---

**Algorithm 3:** Recover a 2D-SPERNER solution from 3-out-of- $k+1$  Approximate SPERNER solutions
 

---

**Input** : vectors  $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \mathbf{x}^{(3)} \in \Delta^k$   
**Output** : vectors  $\hat{\mathbf{x}}^{(1)}, \hat{\mathbf{x}}^{(2)}, \hat{\mathbf{x}}^{(3)} \in \Delta^2$

```

1 for  $i \in \{2, 3, \dots, k-1\}$  do
2   for  $j \in [3]$  do
3     if  $C$  has 3 different colors in  $\mathcal{N}(\mathbf{y}^{(i)}(\mathbf{x}^{(j)}))$  then
4        $\hat{\mathbf{x}}^{(1)}, \hat{\mathbf{x}}^{(2)}, \hat{\mathbf{x}}^{(3)} \leftarrow$  any 3 points colored differently by  $C$  in  $\mathcal{N}(\mathbf{y}^{(i)}(\mathbf{x}^{(j)}))$  // the
5         definition of  $\mathbf{y}^{(i)}(\mathbf{x}^{(j)})$  follows Algorithm 2
6       return  $\hat{\mathbf{x}}^{(1)}, \hat{\mathbf{x}}^{(2)}, \hat{\mathbf{x}}^{(3)}$  // trichromatic triangle found while simulation
7
8 return  $\mathbf{y}^{(k)}(\mathbf{x}^{(1)}), \mathbf{y}^{(k)}(\mathbf{x}^{(2)}), \mathbf{y}^{(k)}(\mathbf{x}^{(3)})$  // trichromatic triangle found when lifting  $C^{(k-1)}$  to  $C^{(k)}$ 

```

---

Our recovery algorithm (formalized in Algorithm 3) simulates Algorithm 2 for each  $\mathbf{x}^{(j)}$ . At time  $i \in [2, k-1]$  when we have computed  $\mathbf{y}^{(i)}(\mathbf{x}^{(j)})$  for each  $j \in [3]$ , we examine whether one of them is inside a trichromatic region, i.e., whether  $C$  has three different colors in  $\mathcal{N}(\mathbf{y}^{(i)}(\mathbf{x}^{(j)}))$  for some  $j \in [3]$ . Because we turn each point in the 2D-rectSPERNER instance to a  $1.6\epsilon \times 1.6\epsilon$  square in the core region, and outside the core region all color switches are defined by  $O(1)$  line segments, we can compute all the connected parts of each color within any  $\mathcal{N}(\mathbf{y})$  in polynomial time. After we finish the simulation, we simply output  $\mathbf{y}^{(k)}(\mathbf{x}^{(1)}), \mathbf{y}^{(k)}(\mathbf{x}^{(2)}), \mathbf{y}^{(k)}(\mathbf{x}^{(3)})$  as a solution for  $C$ .

It is clear that any output during the simulation phase of this recovery algorithm gives a valid solution for the 2D-SPERNER instance  $C$ . To prove Lemma 42, we only need to show that the three intermediate projections after the simulation form a valid solution for  $C$  if we output after the simulation phase. Or equivalently, if each  $\mathbf{y}^{(i)}(\mathbf{x}^{(j)})$  does not lie in a trichromatic region, the final converted coordinates,  $\mathbf{y}^{(k)}(\mathbf{x}^{(1)}), \mathbf{y}^{(k)}(\mathbf{x}^{(2)}), \mathbf{y}^{(k)}(\mathbf{x}^{(3)})$ , give a solution for  $C$ .

Our proof will be considers two cases; the first is where the given solution  $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \mathbf{x}^{(3)}$  for  $C^{(k)}$  further enjoys the following property:

$$\forall 2 \leq i \leq k, \forall j \in [3], \quad P_{i+1}^{(k-i)}(\mathbf{x}^{(j)}) \leq 0.9. \quad (9)$$

One benefit of first considering this case is that we don't have too crazy projections in Algorithm 3, which can help us significantly simplify the proof. The formal intermediate technical result is presented in Corollary 46. Later, in the second step, we will explain how to prove Lemma 42 in the case where this property is not satisfied.

**Case 1: the output satisfies Eq. (9).** The following lemma presents us a Lipschitz property of the projection step for this case.

**Lemma 43.** Consider any  $k \geq 2$ . If  $\mathbf{x}, \mathbf{x}' \in \Delta^k$  satisfy  $x_{k+1}, x'_{k+1} \leq 0.9$ , then we have

$$\|\mathbf{P}(\mathbf{x}) - \mathbf{P}(\mathbf{x}')\|_\infty \leq 110 \cdot \|\mathbf{x} - \mathbf{x}'\|_\infty.$$

*Proof.* For any  $i \in [k]$ , we have

$$\begin{aligned}
|P_i(\mathbf{x}) - P_i(\mathbf{x}')| &= \left| \frac{x_i}{1-x_{k+1}} - \frac{x'_i}{1-x'_{k+1}} \right| \\
&\leq \left| \frac{x_i}{1-x_{k+1}} - \frac{x'_i}{1-x_{k+1}} \right| + \left| \frac{x'_i}{1-x_{k+1}} - \frac{x'_i}{1-x'_{k+1}} \right| \\
&= \left| \frac{x_i - x'_i}{1-x_{k+1}} \right| + |x'_i| \cdot \left| \frac{x_{k+1} - x'_{k+1}}{(1-x_{k+1})(1-x'_{k+1})} \right| \\
&\leq 10 \cdot |x_i - x'_i| + 100 \cdot |x_{k+1} - x'_{k+1}| \quad (x_{k+1}, x'_{k+1} \leq 0.9) \\
&\leq 110 \cdot \|\mathbf{x} - \mathbf{x}'\|_\infty .
\end{aligned}$$

According to the definition of  $\ell_\infty$ -norm, we complete the proof.  $\square$

We can further use this Lipschitzness of the projection step to obtain Lipschitzness for the intermediate projections we use in [Algorithm 2](#).

**Lemma 44.** *Consider any  $k \geq 2$  and any  $\mathbf{x}^{(1)}, \mathbf{x}^{(2)} \in \Delta^k$ . Suppose that  $P_{i+1}^{(k-i)}(\mathbf{x}^{(j)}) \leq 0.9$  for any  $2 \leq i \leq k$  and any  $j \in [2]$ . Also, suppose that  $\mathcal{N}(\mathbf{y}^{(i)}(\mathbf{x}^{(j)}))$  is at most bichromatic for any  $2 \leq i < k$  and any  $j \in [2]$ . Then, for any  $2 \leq i \leq k$ , we have Lipschitzness for the third coordinate of the intermediate projections:*

$$|y_3^{(i)}(\mathbf{x}^{(1)}) - y_3^{(i)}(\mathbf{x}^{(2)})| \leq 2^{O(k)} \cdot \|\mathbf{x}^{(1)} - \mathbf{x}^{(2)}\|_\infty .$$

Furthermore, at least one of the following is satisfied for the second coordinates of the intermediate projections:

- **Lipschitz:**  $|y_2^{(i)}(\mathbf{x}^{(1)}) - y_2^{(i)}(\mathbf{x}^{(2)})| \leq 2^{2in+O(k)} \cdot \|\mathbf{x}^{(1)} - \mathbf{x}^{(2)}\|_\infty$ , or
- **both are on the left/right boundaries:** for any  $j \in [2]$ , we have either  $y_1^{(i)}(\mathbf{x}^{(j)}) = 0$  or  $y_2^{(i)}(\mathbf{x}^{(j)}) = 0$ .

*Proof.* We prove this lemma by induction on  $i$ . The base case is when  $i = 2$ . Its proof is straightforward since we always have  $\mathbf{y}^{(2)}(\mathbf{x}) = \mathbf{P}^{(k-2)}(\mathbf{x})$  and the first bullet is satisfied by the Lipschitzness of projection [Lemma 43](#).

Consider any  $i_0 \geq 3$ . Suppose that we have proved this lemma for  $i < i_0$ . Next, we consider when  $i = i_0$ . Note that we have  $y_3^{(i)}(\mathbf{x}) = P_{i+1}^{(k-i)}(\mathbf{x})$  for any  $\mathbf{x}$ . By the premise that  $P_{i+1}^{(k-i)}(\mathbf{x}^{(j)}) \leq 0.9$ , we can use [Lemma 43](#) to get:

$$\begin{aligned}
\left| y_3^{(i)}(\mathbf{x}^{(1)}) - y_3^{(i)}(\mathbf{x}^{(2)}) \right| &\leq \|\mathbf{P}^{(k-i)}(\mathbf{x}^{(1)}) - \mathbf{P}^{(k-i)}(\mathbf{x}^{(2)})\|_\infty \\
&\leq 110 \cdot \|\mathbf{P}^{(k-i-1)}(\mathbf{x}^{(1)}) - \mathbf{P}^{(k-i-1)}(\mathbf{x}^{(2)})\|_\infty \\
&\leq \dots \\
&\leq 110^{k-i} \cdot \|\mathbf{x}^{(1)} - \mathbf{x}^{(2)}\|_\infty \leq 2^{7k} \cdot \|\mathbf{x}^{(1)} - \mathbf{x}^{(2)}\|_\infty ,
\end{aligned}$$

which establish the first statement of this lemma.

Next, we establish the second statement of this lemma, in which we need to prove at least one of the bullets is satisfied. For the induction hypothesis, we will use the following more specific version of the first bullet

$$|y_2^{(i)}(\mathbf{x}^{(1)}) - y_2^{(i)}(\mathbf{x}^{(2)})| \leq 2^{2in+7k+\log 2i} \cdot \|\mathbf{x}^{(1)} - \mathbf{x}^{(2)}\|_\infty .$$

Note that  $y_2^{(i)}(\mathbf{x}) = (1 - y_3^{(i)}(\mathbf{x})) \cdot \text{rel}^{nn}(\mathbf{y}^{(i-1)}(\mathbf{x}))$ . Then, it is easy to obtain that

$$\begin{aligned} \left| y_2^{(i)}(\mathbf{x}^{(1)}) - y_2^{(i)}(\mathbf{x}^{(2)}) \right| &= \left| (1 - y_3^{(i)}(\mathbf{x}^{(1)})) \cdot \text{rel}^{nn}(\mathbf{y}^{(i-1)}(\mathbf{x}^{(1)})) - (1 - y_3^{(i)}(\mathbf{x}^{(2)})) \cdot \text{rel}^{nn}(\mathbf{y}^{(i-1)}(\mathbf{x}^{(2)})) \right| \\ &\leq \text{rel}^{nn}(\mathbf{y}^{(i-1)}(\mathbf{x}^{(1)})) \cdot \left| y_3^{(i)}(\mathbf{x}^{(1)}) - y_3^{(i)}(\mathbf{x}^{(2)}) \right| \\ &\quad + (1 - y_3^{(i)}(\mathbf{x}^{(2)})) \cdot \left| \text{rel}^{nn}(\mathbf{y}^{(i-1)}(\mathbf{x}^{(1)})) - \text{rel}^{nn}(\mathbf{y}^{(i-1)}(\mathbf{x}^{(2)})) \right| \\ &\leq \left| y_3^{(i)}(\mathbf{x}^{(1)}) - y_3^{(i)}(\mathbf{x}^{(2)}) \right| + \left| \text{rel}^{nn}(\mathbf{y}^{(i-1)}(\mathbf{x}^{(1)})) - \text{rel}^{nn}(\mathbf{y}^{(i-1)}(\mathbf{x}^{(2)})) \right| \\ &\leq 2^{7k} \cdot \|\mathbf{x}^{(1)} - \mathbf{x}^{(2)}\|_\infty + \left| \text{rel}^{nn}(\mathbf{y}^{(i-1)}(\mathbf{x}^{(1)})) - \text{rel}^{nn}(\mathbf{y}^{(i-1)}(\mathbf{x}^{(2)})) \right|. \end{aligned}$$

Suppose that our induction hypothesis gives the first bullet for  $i - 1$ . Combining the Lipschitzness on the third coordinate, we have

$$\|\mathbf{y}^{(i-1)}(\mathbf{x}^{(1)}) - \mathbf{y}^{(i-1)}(\mathbf{x}^{(2)})\|_\infty \leq (2^{2(i-1)n+7k+\log 2(i-1)} + 2^{7k}) \cdot \|\mathbf{x}^{(1)} - \mathbf{x}^{(2)}\|_\infty.$$

If  $\text{rel}^{nn}(\mathbf{y}^{(i-1)}(\mathbf{x}^{(1)})), \text{rel}^{nn}(\mathbf{y}^{(i-1)}(\mathbf{x}^{(2)})) \in \{0, 1\}$ , we have the second bullet for  $i = i_0$ . Otherwise, according to [Lemma 37](#), we have

$$\begin{aligned} \left| y_2^{(i)}(\mathbf{x}^{(1)}) - y_2^{(i)}(\mathbf{x}^{(2)}) \right| &\leq 2^{7k} \cdot \|\mathbf{x}^{(1)} - \mathbf{x}^{(2)}\|_\infty + 2^{2n} \cdot \left| y^{(i-1)}(\mathbf{x}^{(1)}) - y^{(i-1)}(\mathbf{x}^{(2)}) \right| \\ &\leq 2^{7k} \cdot \|\mathbf{x}^{(1)} - \mathbf{x}^{(2)}\|_\infty + 2^{2n} \cdot (2^{2(i-1)n+7k+\log 2(i-1)} + 2^{7k}) \cdot \|\mathbf{x}^{(1)} - \mathbf{x}^{(2)}\|_\infty \\ &\leq 2^{7k} \cdot \|\mathbf{x}^{(1)} - \mathbf{x}^{(2)}\|_\infty + (2i - 1) \cdot 2^{2in+7k} \cdot \|\mathbf{x}^{(1)} - \mathbf{x}^{(2)}\|_\infty \\ &\leq 2^{2in+7k+\log 2i} \cdot \|\mathbf{x}^{(1)} - \mathbf{x}^{(2)}\|_\infty, \end{aligned}$$

which gives the first bullet for  $i = i_0$ .

On the other hand, suppose that the induction hypothesis further gives us the second bullet for  $i - 1$ . That is, for any  $j \in [2]$ , we have

$$\mathbf{y}^{(i-1)}(\mathbf{x}^{(j)}) \in \left\{ \left( 0, 1 - y_3^{(i-1)}(\mathbf{x}^{(j)}), y_3^{(i-1)}(\mathbf{x}^{(j)}) \right), \left( 1 - y_3^{(i-1)}(\mathbf{x}^{(j)}), 0, y_3^{(i-1)}(\mathbf{x}^{(j)}) \right) \right\}.$$

If  $\text{rel}^{nn}(\mathbf{y}^{(i-1)}(\mathbf{x}^{(1)})), \text{rel}^{nn}(\mathbf{y}^{(i-1)}(\mathbf{x}^{(2)})) \in \{0, 1\}$ , we have the second bullet for  $i = i_0$ . Otherwise, according to [Lemma 38](#), we have

$$\begin{aligned} \left| y_2^{(i)}(\mathbf{x}^{(1)}) - y_2^{(i)}(\mathbf{x}^{(2)}) \right| &\leq 2^{7k} \cdot \|\mathbf{x}^{(1)} - \mathbf{x}^{(2)}\|_\infty + 2^{2n} \cdot \left| y_3^{(i-1)}(\mathbf{x}^{(1)}) - y_3^{(i-1)}(\mathbf{x}^{(2)}) \right| \\ &\leq 2^{7k} \cdot \|\mathbf{x}^{(1)} - \mathbf{x}^{(2)}\|_\infty + 2^{2n+7k} \cdot \|\mathbf{x}^{(1)} - \mathbf{x}^{(2)}\|_\infty \\ &\leq 2^{2in+7k+1} \cdot \|\mathbf{x}^{(1)} - \mathbf{x}^{(2)}\|_\infty, \end{aligned}$$

which gives the first bullet for  $i = i_0$ . □

Suppose that [Algorithm 3](#) fails to give us any solution during the simulation phase, i.e., for any  $2 \leq i \leq k - 1$  and  $j \in [3]$ ,  $\mathcal{N}(\mathbf{y}^{(i)}(\mathbf{x}^{(j)}))$  is at most bichromatic. Since the solution of the 3-out-of- $k+1$  Approximate SPERNER instance satisfies  $\|\mathbf{x}^{(j_1)} - \mathbf{x}^{(j_2)}\|_\infty \leq 2^{-3kn}$ , we have  $|y_3^{(i)}(\mathbf{x}^{(j_1)}) - y_3^{(i)}(\mathbf{x}^{(j_2)})| < 2^{-2n} = \varepsilon^2$  for any  $2 \leq i \leq k$  and any  $j_1, j_2 \in [3]$ , and further that

- **the intermediate projections are close to each other:**  $|y_2^{(i)}(\mathbf{x}^{(j_1)}) - y_2^{(i)}(\mathbf{x}^{(j_2)})| < 2^{-2n} = \varepsilon^2$  for any  $j_1, j_2 \in [3]$ , or
- **all intermediate projections are on the left/right boundaries:** for any  $j \in [3]$ , we have either  $y_1^{(i)}(\mathbf{x}^{(j)}) = 0$  or  $y_2^{(i)}(\mathbf{x}^{(j)}) = 0$ .



Next, we can characterize the intermediate palettes used in [Algorithm 2](#) by [Lemma 45](#). The characterization gives equivalence between the set of relevant colors in the palette. In the first case, where the intermediate projections of the three input vectors are close to each other (and at least one of them is not on the left/right boundaries), the palettes are exactly the same. In the second case, where the intermediate projections of the input vectors are all on the left/right boundaries, the only two relevant colors for the points,  $c^{(i)}(\mathbf{x}, i^*(\mathbf{y}^{(i)}(\mathbf{x})))$  and  $c_3^{(i)}(\mathbf{x})$ , are respectively equal.

**Lemma 45.** *Suppose that  $P_{i+1}^{(k-i)}(\mathbf{x}^{(j)}) \leq 0.9$  for any  $2 \leq i \leq k$  and any  $j \in [3]$ . Suppose that  $\mathcal{N}(\mathbf{y}^{(i)}(\mathbf{x}^{(j)}))$  is not trichromatic for any  $2 \leq i < k$  and  $j \in [3]$ . For any  $2 \leq i \leq k$ , at least one of the following holds:*

- *The intermediate projections are close to each other, and we have that the corresponding palettes are the same:  $\mathbf{c}^{(i)}(\mathbf{x}^{(j_1)}) = \mathbf{c}^{(i)}(\mathbf{x}^{(j_2)})$  for any  $j_1, j_2 \in [3]$ .*
- *All intermediate projections are on the left/right boundaries, and the palettes may be different on an irrelevant color, but we still have that both of the following hold:*
  - *The color of the first non-zero coordinate  $c^{(i)}(\mathbf{x}^{(j)}, i^*(\mathbf{y}^{(i)}(\mathbf{x}^{(j)})))$  is the same across all  $j \in [3]$ ; and*
  - *the 3rd color is the same,  $c_3^{(i)}(\mathbf{x}^{(j)}) = i + 1$  for any  $j \in [3]$ .*

Since the output  $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \mathbf{x}^{(3)}$  is trichromatic in  $C^{(k)}$ , we have

$$\left| \{c^{(k)}(\mathbf{x}^{(j)}, C(\mathbf{y}^{(k)}(\mathbf{x}^{(j)})))\}_{j \in [3]} \right| = 3. \quad (10)$$

We should always have the first bullet of [Lemma 45](#) for  $k$  when each  $\mathcal{N}(\mathbf{y}^{(i)}(\mathbf{x}^{(j)}))$  is not trichromatic, because otherwise the second bullet of [Lemma 45](#) violates [Eq. \(10\)](#) as  $C(\mathbf{y}^{(k)}(\mathbf{x}^{(j)})) \in \{i^*(\mathbf{y}^{(k)}(\mathbf{x}^{(j)})), 3\}$ . Note that the first bullet of [Lemma 45](#) and [Eq. \(10\)](#) imply that the colors in the base instance  $C(\mathbf{y}^{(k)}(\mathbf{x}^{(1)}))$ ,  $C(\mathbf{y}^{(k)}(\mathbf{x}^{(2)}))$ , and  $C(\mathbf{y}^{(k)}(\mathbf{x}^{(3)}))$  should be distinct. We can always guarantee that the tuple  $\mathbf{y}^{(k)}(\mathbf{x}^{(1)}), \mathbf{y}^{(k)}(\mathbf{x}^{(2)}), \mathbf{y}^{(k)}(\mathbf{x}^{(3)})$  gives a solution to  $C$ .

**Corollary 46.** *Suppose that  $P_{i+1}^{(k-i)}(\mathbf{x}^{(j)}) \leq 0.9$  for any  $2 \leq i \leq k$  and any  $j \in [3]$ . Suppose that  $\mathcal{N}(\mathbf{y}^{(i)}(\mathbf{x}^{(j)}))$  is not trichromatic for any  $2 \leq i < k$  and  $j \in [3]$ . Then, we have  $|\{C(\mathbf{y}^{(k)}(\mathbf{x}^{(j)})) : j \in [3]\}| = 3$ .*

*Proof of Lemma 45.* We prove this lemma by induction. The base case is when  $i = 2$ . At the beginning of [Algorithm 2](#), we have  $\mathbf{y}^{(2)}(\mathbf{x}^{(j)}) = \mathbf{P}^{(k-2)}(\mathbf{x}^{(j)})$  and  $c^{(2)}(\mathbf{x}^{(j)}) = (1, 2, 3)$  for any  $j \in [3]$ . Because of the Lipschitzness [Lemma 43](#), the first bullet is satisfied.

Consider any  $i_0 \geq 3$ . Assume that we have established this lemma for any  $i = i_0 - 1$ . Next, we establish this lemma for  $i = i_0$ .

First, consider that the first bullet holds for  $i - 1$ . We have  $|y_2^{(i-1)}(\mathbf{x}^{(j_1)}) - y_2^{(i-1)}(\mathbf{x}^{(j_2)})| < 2^{-2n} = \varepsilon^2$  for any  $j_1, j_2 \in [3]$ . Hence,  $d(\mathbf{y}^{(i-1)}(\mathbf{x}^{(j_1)}), \mathbf{y}^{(i-1)}(\mathbf{x}^{(j_2)})) \leq \varepsilon^2$  for any  $j_1, j_2 \in [3]$ . We discuss two cases on whether there is a hot point in  $\mathbf{y}^{(i-1)}(\mathbf{x}^{(1)}), \mathbf{y}^{(i-1)}(\mathbf{x}^{(2)})$  and  $\mathbf{y}^{(i-1)}(\mathbf{x}^{(3)})$ .

- W.l.o.g., suppose that  $\mathbf{y}^{(i-1)}(\mathbf{x}^{(1)})$  is a hot point. According to [Lemma 37](#) and [Lines 4 to 6](#), all the  $i$ -th intermediate projections are close to each other. If there are two different colors in  $\{C(\mathbf{y}^{(i-1)}(\mathbf{x}^{(j)}))\}_{j \in [3]}$ , each  $\mathbf{y}^{(i-1)}(\mathbf{x}^{(j)})$  is hot because we clearly have  $d(\mathbf{y}^{(i-1)}(\mathbf{x}^{(j)}), \mathbf{nn}(\mathbf{y}^{(i-1)}(\mathbf{x}^{(j)}))) < \varepsilon^2$  and [Fact 30](#). Otherwise there is only one color in  $\{C(\mathbf{y}^{(i-1)}(\mathbf{x}^{(j)}))\}_{j \in [3]}$ , since  $d(\cdot, \cdot)$  is a metric, we can obtain the following upper bound on each point's distance to the nearest neighbor by the triangle inequality and [Fact 30](#):

$$\begin{aligned} d(\mathbf{y}^{(i-1)}(\mathbf{x}^{(j)}), \mathbf{nn}(\mathbf{y}^{(i-1)}(\mathbf{x}^{(1)}))) &\leq d(\mathbf{y}^{(i-1)}(\mathbf{x}^{(j)}), \mathbf{y}^{(i-1)}(\mathbf{x}^{(1)})) + d(\mathbf{y}^{(i-1)}(\mathbf{x}^{(1)}), \mathbf{nn}(\mathbf{y}^{(i-1)}(\mathbf{x}^{(1)}))) \\ &< \varepsilon^2 + \varepsilon^2 = 2\varepsilon^2. \end{aligned}$$

Therefore, each  $\mathbf{y}^{(i-1)}(\mathbf{x}^{(j)})$  is either hot or warm by definition, and thus we have  $\hat{C}_{\text{nn}}(\mathbf{y}^{(i-1)}(\mathbf{x}^{(j)})) = C_{\text{nn}}(\mathbf{y}^{(i-1)}(\mathbf{x}^{(j)}))$  for each  $j \in [3]$ . Since they are in a region that is at most bichromatic, the color set  $\{C(\mathbf{y}^{(i-1)}(\mathbf{x}^{(j)})), \hat{C}_{\text{nn}}(\mathbf{y}^{(i-1)}(\mathbf{x}^{(j)}))\}$  is then the same across all  $j \in [3]$ . Since  $c^{(i-1)}(\mathbf{x}^{(1)}) = c^{(i-1)}(\mathbf{x}^{(2)}) = c^{(i-1)}(\mathbf{x}^{(3)})$ , we then have the same  $c^{(i)}(\mathbf{x}^{(j)})$  across all  $j \in [3]$ , which gives the first bullet of this lemma.

- Otherwise, suppose that  $\mathbf{y}^{(i-1)}(\mathbf{x}^{(j)})$  is warm or cold for each  $j \in [3]$ . We have  $\text{rel}^{\text{nn}}(\mathbf{y}^{(i-1)}(\mathbf{x}^{(j)})) \in \{0, 1\}$  for any  $j \in [3]$  in this case. The  $i$ -th intermediate projections are on the left/right boundaries, i.e., we have  $y_1^{(i)}(\mathbf{x}^{(j)}) = 0$  or  $y_2^{(i)}(\mathbf{x}^{(j)}) = 0$  for each  $j \in [3]$ . Note that there is only one color in  $\{C(\mathbf{y}^{(i-1)}(\mathbf{x}^{(j)}))\}_{j \in [3]}$ , because otherwise we clearly have  $d(\mathbf{y}^{(i-1)}(\mathbf{x}^{(j)}), \text{nn}(\mathbf{y}^{(i-1)}(\mathbf{x}^{(j)}))) < \varepsilon^2$  and all points are hot. Note that [Algorithm 2](#) ensures in this scenario that

$$\begin{aligned} i^*(\mathbf{y}^{(i)}(\mathbf{x}^{(j)})) = 1 &\Leftrightarrow \text{rel}^{\text{nn}}(\mathbf{y}^{(i-1)}(\mathbf{x}^{(j)})) = 0 \\ &\Leftrightarrow C(\mathbf{y}^{(i-1)}(\mathbf{x}^{(j)})) < \hat{C}_{\text{nn}}(\mathbf{y}^{(i-1)}(\mathbf{x}^{(j)})) \\ &\Leftrightarrow c^{(i)}(\mathbf{x}^{(j)}, 1) = c^{(i-1)}(\mathbf{x}^{(j)}, C(\mathbf{y}^{(i-1)}(\mathbf{x}^{(j)}))) \end{aligned}$$

where the second *iff* is obtained by [Fact 41](#). We have  $c^{(i)}(\mathbf{x}^{(j)}, i^*(\mathbf{y}^{(i)}(\mathbf{x}^{(j)}))) = c^{(i-1)}(\mathbf{x}^{(j)}, C(\mathbf{y}^{(i-1)}(\mathbf{x}^{(j)})))$  for each  $j \in [3]$ . Since  $c^{(i-1)}(\mathbf{x}^{(1)}) = c^{(i-1)}(\mathbf{x}^{(2)}) = c^{(i-1)}(\mathbf{x}^{(3)})$  and  $C(\mathbf{y}^{(i-1)}(\mathbf{x}^{(1)})) = C(\mathbf{y}^{(i-1)}(\mathbf{x}^{(2)})) = C(\mathbf{y}^{(i-1)}(\mathbf{x}^{(3)}))$ , we have the second bullet for  $i$ .

Second, consider that the second bullet holds for  $i - 1$ . Because  $|y_3^{(i-1)}(\mathbf{x}^{(j_1)}) - y_3^{(i-1)}(\mathbf{x}^{(j_2)})| < \varepsilon^2$  for any  $j_1, j_2 \in [3]$ , according to the characterization of the temperature on the left/right boundaries ([Lemma 34](#)), we have

- for each  $j \in [3]$ ,  $\mathbf{y}^{(i-1)}(\mathbf{x}^{(j)})$  is hot or warm; or
- for each  $j \in [3]$ ,  $\mathbf{y}^{(i-1)}(\mathbf{x}^{(j)})$  is warm or cold.

Since we have  $C(1 - z, 0, z) = 3 = C(0, 1 - z, z)$  or  $C(1 - z, 0, z) = 1, C(0, 1 - z, z) = 2$  for any  $z \in [0, 1]$  ([Fact 26](#)), we have the same  $c^{(i-1)}(\mathbf{x}^{(j)}, C(\mathbf{y}^{(i-1)}(\mathbf{x}^{(j)})))$  across all  $j \in [3]$ . Next, we discuss the above two cases to finish the proof.

- Consider when each  $\mathbf{y}^{(i-1)}(\mathbf{x}^{(j)})$  is hot or warm. We have  $\hat{C}_{\text{nn}}(\mathbf{y}^{(i-1)}(\mathbf{x}^{(j)})) = C_{\text{nn}}(\mathbf{y}^{(i-1)}(\mathbf{x}^{(j)}))$  for each  $j \in [3]$ . Because we have  $C_{\text{nn}}(1 - z, 0, z) = 1, C_{\text{nn}}(0, 1 - z, z) = 2$ , or  $C_{\text{nn}}(1 - z, 0, z) = 3 = C_{\text{nn}}(0, 1 - z, z)$  for each  $z \in [0, 1]$  ([Lemma 34](#)), we have the same  $c^{(i-1)}(\mathbf{x}^{(j)}, \hat{C}_{\text{nn}}(\mathbf{y}^{(i-1)}(\mathbf{x}^{(j)})))$  across all  $j \in [3]$ . Therefore, the palette  $c^{(i)}(\mathbf{x}^{(j)})$  is the same across all  $j \in [3]$ . Because of [Lemma 44](#) and the Lipschitzness of the coordinate converter on the left/right boundaries [Lemma 38](#), the  $i$ -th intermediate projections,  $\mathbf{y}^{(i)}(\mathbf{x}^{(1)}), \mathbf{y}^{(i)}(\mathbf{x}^{(2)})$  and  $\mathbf{y}^{(i)}(\mathbf{x}^{(3)})$ , are close to each other. Hence, we prove the first bullet of this lemma for  $i$ .
- Consider when each  $\mathbf{y}^{(i-1)}(\mathbf{x}^{(j)})$  is warm or cold. We have  $\text{rel}^{\text{nn}}(\mathbf{y}^{(i-1)}(\mathbf{x}^{(j)})) \in \{0, 1\}$  and all  $i$ -th intermediate projections are on the left/right boundaries. Note that there is only one color in  $\{C(\mathbf{y}^{(i-1)}(\mathbf{x}^{(j)}))\}_{j \in [3]}$ , because otherwise we clearly have  $d(\mathbf{y}^{(i-1)}(\mathbf{x}^{(j)}), \text{nn}(\mathbf{y}^{(i-1)}(\mathbf{x}^{(j)}))) < \varepsilon^2$  and all points are hot. According to our earlier discussions, [Algorithm 2](#) ensures in this scenario that  $c^{(i)}(\mathbf{x}^{(j)}, i^*(\mathbf{y}^{(i)}(\mathbf{x}^{(j)}))) = c^{(i-1)}(\mathbf{x}^{(j)}, C(\mathbf{y}^{(i-1)}(\mathbf{x}^{(j)})))$ . Therefore, we have the same  $c^{(i)}(\mathbf{x}^{(j)}, i^*(\mathbf{y}^{(i)}(\mathbf{x}^{(j)})))$  across all  $j \in [3]$ , and thus the second bullet of this lemma holds.  $\square$

**Case 2: the output does not satisfy [Eq. \(9\)](#).** Next, we complete our second step by showing how to prove [Lemma 42](#) without the property ([Eq. \(9\)](#)). Suppose  $\theta^*$  is the minimum threshold  $\theta \geq 2$  such that  $P_{i+1}^{(k-i)}(\mathbf{x}^{(j)}) \leq 0.9$  for any  $\theta \leq i \leq k$  and  $j \in [3]$ . Such threshold always exists because otherwise we have  $x_{k+1}^{(1)}, x_{k+1}^{(2)}, x_{k+1}^{(3)} \geq 0.8$ . This implies  $y_3^{(k)}(\mathbf{x}^{(j)}) > 0.8$  for any  $j \in [3]$ , and according to our construction of the base instance ([Algorithm 1](#)), we have  $C^{(k)}(\mathbf{x}^{(j)}) = c_3^{(k)}(\mathbf{x}^{(j)}) = k + 1$  for any  $j \in [3]$ , violating the

assumption that  $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \mathbf{x}^{(3)}$  form a solution for the 3-out-of- $k+1$  Approximate SPERNER problem. When  $\theta^* = 2$ , it is equivalent with the special case satisfying Eq. (9) and we have proved Lemma 42 for this case. On the other hand, if  $\theta^* > 2$ , we have  $P_{\theta^*}^{(k-\theta^*+1)}(\mathbf{x}^{(j)}) \geq 0.8$  for any  $j \in [3]$  because of Lemma 43 and that  $\|\mathbf{x}^{(i)} - \mathbf{x}^{(j)}\|_\infty \leq 2^{-3kn}$  for any  $i, j \in [3]$ . This means that we have  $C(\mathbf{y}^{(\theta^*-1)}(\mathbf{x}^{(j)})) = 3$ . And because of Fact 39, we have  $\text{rel}^{nn}(\mathbf{y}^{(\theta^*-1)}(\mathbf{x}^{(j)})) = 1$  and  $y_1^{(\theta^*)}(\mathbf{x}^{(j)}) = 0$  for any  $j \in [3]$ . Therefore, running Algorithm 2 on instance  $C^{(k)}$  for  $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \mathbf{x}^{(3)}$  is equivalent to running Algorithm 2 on instance  $C^{(k')}$  with  $k' = k - \theta^* + 2$  for  $\hat{\mathbf{x}}^{(1)}, \hat{\mathbf{x}}^{(2)}, \hat{\mathbf{x}}^{(3)} \in \Delta^{k'}$  such that

$$\forall i \in [k' + 1], j \in [3], \quad \hat{x}_i^{(j)} = \begin{cases} 0 & \text{if } i = 1, \\ 1 - \sum_{i'=\theta^*+1}^{k+1} x_{i'}^{(j)} & \text{if } i = 2, \\ x_{i+\theta^*-2}^{(j)} & \text{if } i > 2. \end{cases}$$

Because this new instance has a smaller number of dimensions and satisfies the condition of our special cases (Eq. (9)), we complete the proof for Lemma 42.

## 7 PPAD-Hardness of Approximate Symmetric Sperner

In this section, we give a more elaborate reduction that guarantees the desired symmetry properties required by Definition 14, and prove the PPAD-completeness of the Approximate Symmetric  $kD$ -SPERNER problem (i.e., our main technical result Theorem 15). We will construct a different chain of instances  $C_{\text{sym}}^{(2)}, C_{\text{sym}}^{(3)}, \dots, C_{\text{sym}}^{(k)}$ . At a high level we make the following two modifications inside the warm-up construction from Section 6: (1) We construct two distinct 2D-SPERNER instances:  $C_{\text{sym}}^{(2)}$  which is used as the basis of our recursive construction, and  $C$  which is used for lifting each  $C_{\text{sym}}^{(i)}$  to  $C_{\text{sym}}^{(i+1)}$ . (2) The actual constructions of 2D-SPERNER instances are more delicate, especially the coordinate converters. As in Section 6, we will use  $2^{-n}$  and  $\varepsilon$  interchangeably in this section. We restate our main technical we will prove as follows.

**Theorem 15.** *For and any  $k = \text{poly}(n)$ , 3-out-of- $k+1$  Approximate Symmetric SPERNER with a triangulation side-length of  $2^{-4kn}$  is PPAD-complete. Further, for any  $k \geq 2$ , it requires a query complexity of  $2^{\Omega(n)}/\text{poly}(n, k)$ .*

As in Section 6, our instances will be constructed in continuous space instead of discrete space. We will establish the following hardness result:

**Theorem 47.** *There exists a chain of functions  $\{C_{\text{sym}}^{(k)} : \Delta^k \rightarrow [k+1]\}_{k \geq 2}$  that satisfy the symmetry defined in Definition 14 and can be computed in  $\text{poly}(|\mathbf{x}|)$  time for each  $\mathbf{x} \in \Delta^k$ , where  $|\mathbf{x}|$  is the bit complexity of  $\mathbf{x}$  and satisfies the following property. For any  $k \geq 2$ , it is PPAD-hard to find three points  $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \mathbf{x}^{(3)} \in \Delta^k$  such that*

- **they are close enough to each other:** for any  $i, j \in [3]$ ,  $\|\mathbf{x}^{(i)} - \mathbf{x}^{(j)}\|_\infty \leq 2^{-4kn}$ ,
- **they induce a trichromatic triangle:**  $|\{C^{(k)}(\mathbf{x}^{(i)}) : i \in [3]\}| = 3$ .

*In particular, if  $C_{\text{sym}}^{(k)}$  is black-box, it requires a query complexity of  $2^{\Omega(n)}/\text{poly}(n, k)$  to find a solution satisfying the above two properties.*

**Remark 48.** *The PPAD-hardness and query complexity lower bound for the continuous 3-out-of- $k+1$  Approximate Symmetric SPERNER instances can be easily generalized for the discrete ones that follow Definition 14. The reduction from the continuous cases to discrete cases is as follows: for any  $\mathbf{x} \in \Delta_{4kn}^k$ , we define*

$$C_{\text{sym,dis}}^{(k)}(\mathbf{x}) := C_{\text{sym}}^{(k)}(\mathbf{x}),$$

*where any trichromatic tuple  $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \mathbf{x}^{(3)}$  in  $C_{\text{sym,dis}}^{(k)}$  is simply a trichromatic tuple in  $C_{\text{sym}}^{(k)}$ .*

Our plan in this section is to prove [Theorem 47](#) and its roadway is as follows. We will first introduce our new coordinate converter in [Subsection 7.1](#). Then, we will (slightly) adjust our previous hard two-dimensional instances  $C^{(2)}$  so that they satisfy the symmetry constraints in [Subsection 7.2](#). Combining this new coordinate converter and these different hard 2D-SPERNER instances with previous recursive construction framework for three or higher dimensions, we have finished all the constructions. We will show the new three or higher dimensional instances (under the new combination) are symmetric and still PPAD-hard in [Subsection 7.3](#). Finally, we will establish the query complexity of the instances in [Subsection 7.4](#). In the entire section, we assume that the base instance  $C$  is fixed and any function in consideration will have oracle access to  $C$ .

## 7.1 A more elaborate coordinate converter

In this subsection, we provide a more elaborate coordinate converter. Recall that in our hard instances for the 3-out-of- $k+1$  Approximate (Unconstrained) SPERNER problem, we define the coordinate converters simply via each point's  $\ell_\infty$ -nearest neighbor with a different color. However, using this coordinate converter, we cannot guarantee the desired symmetry ([Definition 14](#)) even in the 3D-instances. Consider the following two facets:  $\{(x, y, z, 0) : x + y + z = 1\}$  and  $\{(x, y, 0, z) : x + y + z = 1\}$ . In our construction in [Section 6](#), we let  $C^{(3)}(x, y, z, 0)$  be the  $C(x, y, z)$ -th color in the palette  $(1, 2, 3)$ . In contrast, for  $C^{(3)}(x, y, 0, z)$ , we use the  $C((1-z) \cdot (1-\tilde{y}), (1-z) \cdot \tilde{y}, z)$ -th color in the palette  $(1, 2, 4)$ , where  $\tilde{y} = \text{rel}^{nn}(\frac{x}{1-z}, \frac{y}{1-z}, 0)$ . To ensure the symmetry between  $C^{(3)}(x, y, z, 0)$  and  $C^{(3)}(x, y, 0, z)$ , one natural way is to guarantee that we use the same point in the 2D-SPERNER instance for  $(x, y, z, 0)$  and  $(x, y, 0, z)$ , i.e.,

$$y = (1-z) \cdot \tilde{y} = (1-z) \cdot \text{rel}^{nn} \left( 1 - \frac{y}{1-z}, \frac{y}{1-z}, 0 \right),$$

which can hold if the coordinate converter is the identity on the bottom 1-simplex, i.e.,  $\text{rel}^{nn}(1-y, y, 0) = y$ .

Therefore, our primary motivation for using a more elaborate coordinate converter is to ensure that it gives an identity mapping on the bottom 1-simplex. We also have to carefully interpolate between this boundary constraint and maintaining the desiderata of the existing construction on the interior.

**Interpolation.** To interpolate the coordinate converter, we define the following *shrinking factor* for the distance, which depends only on the third coordinate of each point and will be applied on the second coordinate when computing the distance. The formula [Eq. \(11\)](#) of this shrinking factor can be viewed as an interpolation between  $z = 0$  and  $z = 0.05$  using exponential functions so that we will shrink distances of points with  $z = 0$  by a small factor of  $2\epsilon^2$ . We do not shrink the distances (i.e.,  $\alpha(\cdot) = 1$ ) for those with  $z \geq 0.05$ . See [Lemma 49](#) for the details.

$$\alpha(z) = \exp \left( -20(2n-1) \ln 2 \cdot (0.05-z)_+ \right). \quad (11)$$

The intuition of our new definition is that with the distance shrunk, the hot regions expand. Therefore, the above issue of asymmetry between the bottom 2-simplex and the other three 2-simplices in the 3D-instances could be resolved. The key properties of this new coordinate converter towards proving the symmetry are concluded in the following lemma.

**Lemma 49.** *The shrinking factor  $\alpha$  satisfies:*

1.  $\alpha(0) = 2^{-2n+1} = 2\epsilon^2$ .
2. For any  $z \geq 0.05$ ,  $\alpha(z) = 1$ .
3. Lipschitz: for any  $z, z' \in [0, 1]$  such that  $|z - z'| \leq \epsilon$ ,  $|\alpha(z) - \alpha(z')| \leq O(n) \cdot |z - z'|$ .

*Proof.* The first two bullets can be easily obtained via simple calculations. Further, because of the second bullet, it remains to show the third bullet for  $0 \leq z \leq z' \leq 0.05$  such that  $|z - z'| \leq 2^{-\Omega(n)}$ :

$$\begin{aligned}
|\alpha(z) - \alpha(z')| &= \left| \exp(-20(2n-1)\ln 2 \cdot (0.05 - z)) - \exp(-20(2n-1)\ln 2 \cdot (0.05 - z')) \right| \\
&= \left| \exp(-20(2n-1)\ln 2 \cdot (0.05 - z)) \cdot (1 - \exp(20(2n-1)\ln 2 \cdot (z' - z))) \right| \\
&\leq \exp(20(2n-1)\ln 2 \cdot (z' - z)) - 1 \\
&\leq 20(2n-1)\ln 2 \cdot (z' - z) + O(n^2(z' - z)^2) \\
&= O(n) \cdot |z - z'|. \quad \square
\end{aligned}$$

Now we're ready to define the following (asymmetric) *shrunk quasimetric* based on the shrinking factor,

$$d^\alpha(\mathbf{x}, \mathbf{x}') = \max \{ \alpha(x_3) \cdot |x_2 - x'_2|, |x_3 - x'_3| \} \quad (12)$$

**The coordinate converter.** The definition of the coordinate converter follows the previous section's steps. More specifically, for each point  $\mathbf{x} \in \Delta^2$ , we find the infimum shrunk distance from  $\mathbf{x}$  to all points with a fixed color  $c \neq C(\mathbf{x})$ :

$$d_{\min}^\alpha(\mathbf{x}, c) = \inf_{\mathbf{y} \in \Delta^2: C(\mathbf{y})=c} d^\alpha(\mathbf{x}, \mathbf{y}).$$

Then, our definitions of the neighboring color and the nearest neighbor are modified accordingly as follows:

$$\begin{aligned}
C_{\text{nn}}^\alpha(\mathbf{x}) &= \arg \min_{c \in [3]: c \neq C(\mathbf{x})} (d_{\min}^\alpha(\mathbf{x}, c), c), \\
\mathbf{nn}^\alpha(\mathbf{x}) &= \arg \inf_{\mathbf{y} \in \Delta^2: C(\mathbf{y})=C_{\text{nn}}^\alpha(\mathbf{x})} d^\alpha(\mathbf{x}, \mathbf{y}).
\end{aligned} \quad (13)$$

We will continue to use the notions of *hot/warm/cold regions*, where the definitions are nearly the same as in [Section 6](#) (replacing  $d$  by  $d^\alpha$ ).

- The *hot* region consists of points having a shrunk distance to the nearest neighbor strictly less than  $\varepsilon^2$ , i.e., points  $\mathbf{x}$  with  $d^\alpha(\mathbf{x}, \mathbf{nn}^\alpha(\mathbf{x})) < \varepsilon^2$ ; we say that points in this region are *hot*.
- The *warm* region consists of points having a shrunk distance to the nearest neighbor between  $\varepsilon^2$  (inclusive) and  $2\varepsilon^2$  (exclusive), i.e., points  $\mathbf{x}$  with  $d^\alpha(\mathbf{x}, \mathbf{nn}^\alpha(\mathbf{x})) \in [\varepsilon^2, 2\varepsilon^2)$ ; we say that points in this region are *warm*.
- The *cold* region consists of points having a shrunk distance to the nearest neighbor no less than  $2\varepsilon^2$ , i.e., points  $\mathbf{x}$  with  $d^\alpha(\mathbf{x}, \mathbf{nn}^\alpha(\mathbf{x})) \geq 2\varepsilon^2$ ; we say that points in this region are *cold*.

Finally, the new coordinate converter is in the form as the previous one [Eq. \(5\)](#) except that we use everything with a shrinking factor.

$$\text{rel}^{\mathbf{nn}, \alpha}(\mathbf{x}) = \begin{cases} (0.5 - 0.5\varepsilon^{-2} \cdot d^\alpha(\mathbf{x}, \mathbf{nn}^\alpha(\mathbf{x})))_+ & \text{if } \mathbf{x} \text{ is } \textit{hot/warm} \text{ and } C_{\text{nn}}^\alpha(\mathbf{x}) > C(\mathbf{x}), \\ (0.5 + 0.5\varepsilon^{-2} \cdot d^\alpha(\mathbf{x}, \mathbf{nn}^\alpha(\mathbf{x})))_- & \text{if } \mathbf{x} \text{ is } \textit{hot/warm} \text{ and } C_{\text{nn}}^\alpha(\mathbf{x}) < C(\mathbf{x}), \\ 0 & \text{if } \mathbf{x} \text{ is } \textit{cold} \text{ and } C(\mathbf{x}) = 1, \\ 1 & \text{if } \mathbf{x} \text{ is } \textit{cold} \text{ and } C(\mathbf{x}) \in \{2, 3\}. \end{cases} \quad (14)$$

Because we have  $\alpha(x_3) = 1$  for any  $\mathbf{x}$  such that  $x_3 \geq 0.05$ , it is easy to observe that the coordinate converter, along with every intermediate function for its definition, does not change for those points. This observation will help us continue to use a huge fraction of the properties we have established in [Section 6](#).

**Observation 50.** Consider any  $\mathbf{x} \in \Delta^2$  such that  $x_3 \geq 0.05$ . We have

- $d^\alpha(\mathbf{x}, \mathbf{x}') = d(\mathbf{x}, \mathbf{x}')$  for any  $\mathbf{x}' \in \Delta^2$ ;
- $C_{\text{nn}}^\alpha(\mathbf{x}) = C_{\text{nn}}(\mathbf{x})$ ;
- $\mathbf{nn}^\alpha(\mathbf{x}) = \mathbf{nn}(\mathbf{x})$ ;
- $\text{rel}^{\text{nn},\alpha}(\mathbf{x}) = \text{rel}^{\text{nn}}(\mathbf{x})$ .

### 7.1.1 Key properties

Next, we establish the key properties of this new coordinate converter on this family of instances. The only new property we will establish is that this new function is an identity on the base 1-simplex. In addition, we will prove the analogs of all the properties we use in [Section 6 \(Lemmas 33, 34 and 37 and Fact 39\)](#) for this new coordinate converter.

**Property I: identity on the base** With the new definition of the coordinate converter, we can prove that the coordinate converter is simply an identity function for points on the base 1-simplex. Moreover, we characterize the neighbor coloring on the base 1-simplex.

**Lemma 51.** Points on the segment  $(1 - x_2, x_2, 0)$  (for  $x_2 \in (0, 1)$ ) are hot. More specifically, for any  $x_2 \in [0, 1]$ , we have  $\text{rel}^{\text{nn},\alpha}(1 - x_2, x_2, 0) = x_2$ , and

$$C_{\text{nn}}^\alpha(1 - x_2, x_2, 0) = \begin{cases} 1 & \text{if } x_2 > 0.5, \\ 2 & \text{if } x_2 \leq 0.5. \end{cases}$$

*Proof.* Let  $\mathbf{x} = (1 - x_2, x_2, 0)$ . According to [Lemma 49](#) and [Eq. \(12\)](#), we have

$$d^\alpha((1 - x_2, x_2, 0), \mathbf{x}') = \max \{2\varepsilon^2 \cdot |x_2 - x'_2|, |x'_3|\}.$$

Note that in our base instance  $C$ , we have

$$\forall \mathbf{x}' \in \Delta^2, \quad \begin{cases} C(\mathbf{x}') = 1 & \text{if } x'_3 \leq 0.1, x'_2 \leq 0.5, \\ C(\mathbf{x}') = 2 & \text{if } x'_3 \leq 0.1, x'_2 > 0.5. \end{cases}$$

If we have  $x_2 \leq 0.5$  here, any point  $\mathbf{x}'$  with a different color with  $C(\mathbf{x}) = 1$  has either  $x'_2 > 0.5$  or  $x'_3 > 0.1$ . For the second condition ( $x'_3 > 0.1$ ), we have  $d^\alpha(\mathbf{x}, \mathbf{x}') > 0.1 > 2\varepsilon^2$ . On the other hand, it is easy to observe that  $(0.5, 0.5, 0)$  has the minimum distance  $d^\alpha(\mathbf{x}, \mathbf{x}')$  among those points with  $x'_3 \leq 0.1, x'_2 \geq 0.5$ . Therefore,  $\mathbf{nn}^\alpha(\mathbf{x}) = (0.5, 0.5, 0)$  and  $C_{\text{nn}}^\alpha(\mathbf{x}) = 2$ . In particular, according to [Lemma 49](#), we have  $d^\alpha(\mathbf{x}, (0.5, 0.5, 0)) = 2\varepsilon^2 \cdot (0.5 - x_2)$ . According to [Eq. \(14\)](#), because  $C_{\text{nn}}^\alpha(\mathbf{x}) > C(\mathbf{x})$ ,

$$\begin{aligned} \text{rel}^{\text{nn},\alpha}(\mathbf{x}) &= (0.5 - 0.5\varepsilon^{-2} \cdot d^\alpha(\mathbf{x}, \mathbf{nn}^\alpha(\mathbf{x})))_+ \\ &= (0.5 - 0.5\varepsilon^{-2} \cdot d^\alpha(\mathbf{x}, (0.5, 0.5, 0)))_+ \\ &= (0.5 - (0.5 - x_2)) = x_2. \end{aligned}$$

Hence, when  $x_2 \in (0, 0.5]$ ,  $\mathbf{x}$  is in the hot region.

Similarly, if we have  $x_2 > 0.5$  here, we have  $C(\mathbf{x}) = 2$ ,  $\mathbf{nn}^\alpha(\mathbf{x}) = (0.5, 0.5, 0)$  and  $C_{\text{nn}}^\alpha(\mathbf{x}) = 1$ . According to [Lemma 49](#), we have  $d^\alpha(\mathbf{x}, (0.5, 0.5, 0)) = 2\varepsilon^2 \cdot (x_2 - 0.5)$ . According to [Eq. \(14\)](#), because  $C_{\text{nn}}^\alpha(\mathbf{x}) < C(\mathbf{x})$ ,

$$\begin{aligned} \text{rel}^{\text{nn},\alpha}(\mathbf{x}) &= \min \{0.5 + 0.5\varepsilon^{-2} \cdot d^\alpha(\mathbf{x}, \mathbf{nn}^\alpha(\mathbf{x})), 1\} \\ &= \min \{0.5 + 0.5\varepsilon^{-2} \cdot d^\alpha(\mathbf{x}, (0.5, 0.5, 0)), 1\} \\ &= \min \{0.5 + (x_2 - 0.5), 1\} = x_2. \end{aligned}$$

Hence, when  $x_2 \in (0.5, 1)$ ,  $\mathbf{x}$  is in the hot region.

In conclusion, for any  $x_2 \in (0, 1)$ , the point  $\mathbf{x}$  is in the hot region.  $\square$

**Property II: Polynomial time computation** As in [Section 6](#), we show that we can always output the true converted coordinate  $\text{rel}^{nn,\alpha}(\mathbf{x})$  and, whenever  $\mathbf{x}$  is hot or warm (i.e.,  $d^\alpha(\mathbf{x}, \mathbf{nn}^\alpha(\mathbf{x})) < 2\varepsilon^2$ ), we can also output the neighboring color  $C_{nn}^\alpha(\mathbf{x})$ .

**Lemma 52.** *Given oracle access to  $C$ , there is a polynomial-time algorithm that takes any  $\mathbf{x} \in \Delta^2$  as input and that outputs  $\text{rel}^{nn,\alpha}(\mathbf{x})$ . Furthermore, if  $d^\alpha(\mathbf{x}, \mathbf{nn}^\alpha(\mathbf{x})) < 2\varepsilon^2$ , the algorithm can also compute  $C_{nn}^\alpha(\mathbf{x})$  in polynomial time.*

*Proof.* Note that the converted coordinates are the same for any  $\mathbf{x}$  with  $x_3 \geq 0.05$  ([Observation 50](#)), we only need to show for  $x_3 \leq 0.05$  because we have established [Lemma 33](#). W.l.o.g., we suppose that the input  $\mathbf{x} \in \Delta^2$  satisfies  $x_3 \leq 0.05$ .

As in the proof of [Lemma 33](#), it suffices to show how to compute  $d^\alpha(\mathbf{x}, \mathbf{nn}^\alpha(\mathbf{x}))$  and  $C_{nn}^\alpha(\mathbf{x})$  when  $\mathbf{x}$  is hot or warm. Note that in our base instance  $C$ , we have

$$\forall \mathbf{x}' \in \Delta^2, \quad \begin{cases} C(\mathbf{x}') = 1 & \text{if } x'_3 \leq 0.1, x'_2 \leq 0.5, \\ C(\mathbf{x}') = 2 & \text{if } x'_3 \leq 0.1, x'_2 > 0.5. \end{cases}$$

Also, note that our shrinking distance can be lower bounded by the difference on the third coordinate:

$$d^\alpha(\mathbf{x}, \mathbf{x}') = \max \{ \alpha(x_3) \cdot |x_2 - x'_2|, |x_3 - x'_3| \} \geq |x_3 - x'_3|.$$

We have  $d^\alpha(\mathbf{x}, \mathbf{nn}^\alpha(\mathbf{x})) < 2\varepsilon^2$  only when  $\mathbf{nn}_3^\alpha(\mathbf{x}) \leq 0.05 + 2\varepsilon^2 < 0.1$ . Among points  $\mathbf{x}'$  such that  $x'_3 < 0.1$ , the nearest<sup>9</sup> differently colored point of  $\mathbf{x}$  is clearly  $(0.5 - x_3, 0.5, x_3)$ . Therefore, when  $\mathbf{x}$  is hot or warm, we have  $\mathbf{nn}^\alpha(\mathbf{x}) = (0.5 - x_3, 0.5, x_3)$ . Because of this simple characterization of  $\mathbf{nn}^\alpha(\cdot)$  for hot and warm points, we can compute  $d^\alpha(\mathbf{x}, (0.5 - x_3, 0.5, x_3))$  to decide if  $\mathbf{x}$  is hot or warm and then compute the value of  $d^\alpha(\mathbf{x}, \mathbf{nn}^\alpha(\mathbf{x}))$ . In addition, for hot and warm points,  $C_{nn}^\alpha(\mathbf{x})$  equals the color in  $\{1, 2\}$  that does not equal  $C(\mathbf{x})$ .  $C_{nn}^\alpha(\mathbf{x})$  is also easy to compute when  $\mathbf{x}$  is hot or warm.  $\square$

**Property III: Symmetry** Recall [Lemma 34](#), where we give a complete characterization of the temperature of each point on the line segments  $(1 - x_3, 0, x_3)$  and  $(0, 1 - x_3, x_3)$ . In the complete characterization, the set of hot and warm points consists of those with  $x_3 \in 0.1 \pm 2\varepsilon^2$ . Here, we generalize it to a (slightly) incomplete characterization under the new coordinate converter, where  $\mathbf{x}$  has the same characterization as in [Section 6](#) if  $x_3 \in 0.1 \pm 2\varepsilon^2$ , and  $\mathbf{x}$  is not hot otherwise.

**Lemma 53.** *For points on the line segments  $(1 - x_3, 0, x_3)$  and  $(0, 1 - x_3, x_3)$ , we have*

- if  $x_3 \notin 0.1 \pm 2\varepsilon^2$ , then  $(1 - x_3, 0, x_3)$  and  $(0, 1 - x_3, x_3)$  are either cold or warm;
- otherwise, if  $x_3 \in 0.1 \pm 2\varepsilon^2$ , then

$$\text{rel}^{nn,\alpha}(1 - x_3, 0, x_3) = \left( 1/2 + \frac{x_3 - 0.1}{2\varepsilon^2} \right)_{[0,1]} = \text{rel}^{nn,\alpha}(0, 1 - x_3, x_3), \quad (15)$$

and the neighboring color is characterized as follows

$$C_{nn}^\alpha(1 - x_3, 0, x_3) = \begin{cases} 3 & \text{if } x_3 \leq 0.1, \\ 1 & \text{if } x_3 > 0.1, \end{cases} \quad \text{and} \quad C_{nn}^\alpha(0, 1 - x_3, x_3) = \begin{cases} 3 & \text{if } x_3 \leq 0.1, \\ 2 & \text{if } x_3 > 0.1. \end{cases}$$

<sup>9</sup>Here, it may be possible that the nearest differently colored point of  $\mathbf{x}$  is given by the limit of a sequence of infinite many points that have different colors with  $\mathbf{x}$ .

In particular, we have  $\text{rel}^{nn,\alpha}(1 - x_3, 0, x_3) \sim_{\text{rel}} \text{rel}^{nn,\alpha}(0, 1 - x_3, x_3) \sim_{\text{rel}} (0.5 + 0.5\varepsilon^{-2} \cdot (x_3 - 0.1))_{[0,1]}$  for any  $x_3 \in [0, 1]$ .

*Proof.* Because of [Observation 50](#), the characterization is the same for any  $z \geq 0.05$ , we only need to show for  $z \leq 0.05$  because we have established [Lemma 34](#). That is, we want to show that for any  $x_3 \leq 0.05$ ,

$$\text{rel}^{nn,\alpha}(1 - x_3, 0, x_3), \text{rel}^{nn,\alpha}(0, 1 - x_3, x_3) \in \{0, 1\} .$$

According to our new coordinate converter [Eq. \(14\)](#), it suffices to show that for any  $x_3 \leq 0.05$  and any  $\mathbf{x} \in \{(1 - x_3, 0, x_3), (0, 1 - x_3, x_3)\}$ ,

$$d^\alpha(\mathbf{x}, \mathbf{nn}^\alpha(\mathbf{x})) \geq \varepsilon^2 . \quad (16)$$

Note that in our base instance  $C$ , we have

$$\forall \mathbf{x}' \in \Delta^2, \quad \begin{cases} C(\mathbf{x}') = 1 & \text{if } x'_3 \leq 0.1, x'_2 \leq 0.5 , \\ C(\mathbf{x}') = 2 & \text{if } x'_3 \leq 0.1, x'_2 > 0.5 . \end{cases}$$

First, we consider the case when  $\mathbf{x} = (1 - x_3, 0, x_3)$  for  $x_3 \leq 0.05$ . We have  $C(\mathbf{x}) = 1$ , and for any  $\mathbf{x}' \in \Delta^2$ ,  $C(\mathbf{x}) \neq C(\mathbf{x}')$  only if  $x'_2 \geq 0.5$  or  $x'_3 \geq 0.1$ . Since  $d^\alpha(\mathbf{x}, \mathbf{x}') \geq |x_3 - x'_3|$ , if  $x'_3 \geq 0.1$ , we have  $d^\alpha(\mathbf{x}, \mathbf{x}') \geq 0.05 > \varepsilon^2$ . On the other hand, if  $x'_2 \geq 0.5$ , because of [Lemma 49](#), we have

$$d^\alpha(\mathbf{x}, \mathbf{x}') \geq \alpha(x_3) \cdot |0 - x'_2| \geq \alpha(0) \cdot 0.5 = \varepsilon^2 .$$

Therefore, we have  $d^\alpha(\mathbf{x}, \mathbf{x}')$  for any  $C(\mathbf{x}') \neq C(\mathbf{x})$ , and thus we have proved [Eq. \(16\)](#) for this case.

Second, we consider the case when  $\mathbf{x} = (0, 1 - x_3, x_3)$  for  $x_3 \leq 0.05$ . We have  $C(\mathbf{x}) = 2$ , and for any  $\mathbf{x}' \in \Delta^2$ ,  $C(\mathbf{x}) \neq C(\mathbf{x}')$  only if  $x'_2 \leq 0.5$  or  $x'_3 \geq 0.1$ . Since  $d^\alpha(\mathbf{x}, \mathbf{x}') \geq |x_3 - x'_3|$ , if  $x'_3 \geq 0.1$ , we have  $d^\alpha(\mathbf{x}, \mathbf{x}') \geq 0.05 > \varepsilon^2$ . On the other hand, if  $x'_2 \leq 0.5$ , because of [Lemma 49](#), we have

$$\begin{aligned} d^\alpha(\mathbf{x}, \mathbf{x}') &= \alpha(x_3) \cdot |(1 - x_3) - x'_2| \\ &\geq \alpha(x_3) \cdot (0.5 - x_3) \\ &= \exp(-20(2n - 1) \ln 2 \cdot (0.05 - x_3)) \cdot (0.5 - x_3) \\ &= 2^{-2n+1} \cdot \exp(20(2n - 1) \ln 2 \cdot x_3) \cdot (0.5 - x_3) \\ &\geq 2^{-2n+1} \cdot (1 + 20(2n - 1) \ln 2 \cdot x_3) \cdot (0.5 - x_3) \\ &\geq 2^{-2n+1} \cdot 0.5 = 2^{-2n} = \varepsilon^2 \end{aligned} \quad (x_3 \leq 0.05)$$

Hence, we complete the proof. □

**Property IV: Lipschitzness** We prove that the new coordinate converter is Lipschitz in the same sense as [Lemma 37](#), but with a slightly larger Lipschitz factor.

**Lemma 54.** For any point  $\mathbf{x}, \mathbf{x}'$ , at least one of the following properties is satisfied:

- **one is in a trichromatic region:** there are three different colors in  $\mathcal{N}(\mathbf{x})$  or  $\mathcal{N}(\mathbf{x}')$ ;
- **Lipschitz in the hot&warm regions:**  $|\text{rel}^{nn,\alpha}(\mathbf{x}) - \text{rel}^{nn,\alpha}(\mathbf{x}')| \leq \varepsilon^{-3} \cdot \|\mathbf{x} - \mathbf{x}'\|_\infty$ ;
- **both are in the warm&cold regions:**  $\text{rel}^{nn,\alpha}(\mathbf{x}), \text{rel}^{nn,\alpha}(\mathbf{x}') \in \{0, 1\}$ ;



*Proof.* Because of [Observation 50](#) and [Lemma 37](#), we have established this lemma if  $x_3, x'_3 \geq 0.05$ . If  $|x'_3 - x_3| \geq \varepsilon^3$ , the second property trivially holds. Therefore, it suffices to establish this lemma for the case where  $x_3, x'_3 \leq 0.06$  to finish its proof.

We will use the following fact about the shrunk distance to each point's nearest neighbor and the new coordinate converter, where the proof is based on simple calculations and deferred to [Appendix A.3](#).

**Fact 55.** Consider an auxillary function  $g(\mathbf{y}) = \alpha(y_3) \cdot (y_2 - 0.5)$ . For any  $\mathbf{y} \in \Delta^2$  such that  $y_3 \leq 0.06$ , warm and hot points satisfies  $g(\mathbf{y}) \in \pm 2\varepsilon^2$ . Furthermore,

$$d^\alpha(\mathbf{y}, \mathbf{nn}^\alpha(\mathbf{y})) = |g(\mathbf{y})|, \quad \text{and} \quad \text{rel}^{\mathbf{nn}, \alpha}(\mathbf{y}) = (0.5 + 0.5\varepsilon^{-2} \cdot g(\mathbf{y}))_{[0,1]}.$$

Similarly as in the proof of [Lemma 37](#), we only need to show that the Lipschitz property,  $|\text{rel}^{\mathbf{nn}, \alpha}(\mathbf{x}) - \text{rel}^{\mathbf{nn}, \alpha}(\mathbf{x}')| \leq \varepsilon^{-3} \cdot \|\mathbf{x} - \mathbf{x}'\|_\infty$ , under the following two assumptions<sup>10</sup>: (1)  $\|\mathbf{x} - \mathbf{x}'\|_\infty \leq \varepsilon^3$ ; and (2)  $\mathbf{x}$  is hot, i.e.,  $\text{rel}^{\mathbf{nn}, \alpha}(\mathbf{x}) \in (0, 1)$ .

Under the first assumption, we have

$$\begin{aligned} |g(\mathbf{x}) - g(\mathbf{x}')| &= \left| \alpha(x_3) \cdot (x_2 - 0.5) - \alpha(x'_3) \cdot (x'_2 - 0.5) \right| \\ &\leq \alpha(x_3) \cdot |x_2 - x'_2| + \left| \alpha(x_3) - \alpha(x'_3) \right| \cdot |x'_2 - 0.5| \\ &\leq |x_2 - x'_2| + O(n) \cdot |x_3 - x'_3| \quad (\alpha \text{ is increasing and } \text{Lemma 49}) \\ &\leq O(n) \cdot \|\mathbf{x} - \mathbf{x}'\|_\infty. \quad (\text{Lemma 49}) \end{aligned}$$

First, we show that  $\mathbf{x}'$  is hot or warm. According to [Fact 55](#), because  $\mathbf{x}$  is hot and  $0.5 + 0.5\varepsilon^{-2} \cdot g(\mathbf{x}) \in (0, 1)$  only when  $g(\mathbf{x}) \in \varepsilon^2$ ,  $g(\mathbf{x}') \leq O(n)\|\mathbf{x} - \mathbf{x}'\|_\infty + g(\mathbf{x}) < O(n) \cdot \varepsilon^3 + \varepsilon^2 < 2\varepsilon^2$ . Then, we establish the Lipschitz property for  $\mathbf{x}$  and  $\mathbf{x}'$  under the two assumptions. Because of [Fact 55](#) and the fact that  $|(a)_{[0,1]} - (b)_{[0,1]}| \leq |a - b|$ , we have

$$\begin{aligned} |\text{rel}^{\mathbf{nn}, \alpha}(\mathbf{x}) - \text{rel}^{\mathbf{nn}, \alpha}(\mathbf{x}')| &\leq \left| 0.5\varepsilon^{-2} \cdot g(\mathbf{x}) - 0.5\varepsilon^{-2} \cdot g(\mathbf{x}') \right| \\ &= 0.5\varepsilon^{-2} \cdot |g(\mathbf{x}) - g(\mathbf{x}')| \leq \varepsilon^{-3} \cdot \|\mathbf{x} - \mathbf{x}'\|_\infty. \quad \square \end{aligned}$$

Similar as [Lemma 38](#), we can prove Lipschitzness for the converted coordinates on the left/right boundaries by [Lemma 53](#).

**Lemma 56.** For any  $z, z' \in [0, 1]$  and any pair of points  $\mathbf{x}, \mathbf{x}'$  such that  $\mathbf{x} \in \{(0, 1 - z, z), (1 - z, 0, z)\}$  and  $\mathbf{x}' \in \{(0, 1 - z', z'), (1 - z', 0, z')\}$ , at least one of the following properties is satisfied:

- **Lipschitz in the hot&warm regions:**  $|\text{rel}^{\mathbf{nn}, \alpha}(\mathbf{x}) - \text{rel}^{\mathbf{nn}, \alpha}(\mathbf{x}')| \leq \varepsilon^{-2} \cdot |z - z'|$ ;
- **both are in the warm&cold regions:**  $\text{rel}^{\mathbf{nn}, \alpha}(\mathbf{x}), \text{rel}^{\mathbf{nn}, \alpha}(\mathbf{x}') \in \{0, 1\}$ .

**Property V: Cold on the top** Finally, we prove that the converted coordinates are trivial for the points with reasonably high values on the third dimension ( $x_3 > 0.5$ ). Because  $\text{rel}^{\mathbf{nn}}(\mathbf{x}) = \text{rel}^{\mathbf{nn}, \alpha}(\mathbf{x})$  for those  $x_3 > 0.5$ , this fact holds trivially for the new coordinate converter because of our earlier [Fact 39](#).

**Fact 57.** For any  $\mathbf{x} \in \Delta^2$  such that  $x_3 \geq 0.5$ , we have  $\text{rel}^{\mathbf{nn}, \alpha}(\mathbf{x}) = 1$ .

<sup>10</sup>We don't need the earlier assumption that  $\mathcal{N}(\mathbf{x})$  and  $\mathcal{N}(\mathbf{x}')$  are both at most bichromatic here. This is because [Fact 28](#) ensures that  $\mathbf{x}, \mathbf{x}'$  here are not in a trichromatic region.

---

**Algorithm 4:** 3-out-of- $k+1$  Approximate Symmetric SPERNER Instance  $C_{\text{sym}}^{(k)}(\mathbf{x})$ 


---

**Input** : vector  $\mathbf{x} \in \Delta^k$   
**Output:** color  $c \in [k+1]$

- 1  $\varepsilon \leftarrow 2^{-n}$
- 2  $y_0 \leftarrow P_2^{(k-1)}(\mathbf{x})$
- 3  $\tilde{y}_0 \leftarrow (0.5 + 0.5\varepsilon^{-2} \cdot (y_0 - 0.1))_{[0,1]}$  // convert  $\mathbf{x}$ 's projection to the base 1-simplex
- 4  $z \leftarrow P_3^{(k-2)}(\mathbf{x})$
- 5  $\mathbf{y}^{(2)} \leftarrow ((1-z) \cdot (1-\tilde{y}_0), (1-z) \cdot \tilde{y}_0, z)$  // initiate  $\mathbf{y}$  by the conversion and  $\mathbf{x}$ 's projection to the base 2-simplex
- 6  $\mathbf{c}^{(2)} \leftarrow (1, 2, 3)$  // initiate the set of 3 colors
- 7 **for**  $i \in \{3, \dots, k\}$  **do**
- 8      $y_3^{(i)} \leftarrow P_{i+1}^{(k-i)}(\mathbf{x})$  // find the new z-coordinate from the projection of  $\mathbf{x}$
- 9      $y_2^{(i)} \leftarrow (1 - y_3^{(i)}) \cdot \text{rel}^{\text{nn}, \alpha}(\mathbf{y}^{(i-1)})$
- 10     $y_1^{(i)} \leftarrow (1 - y_3^{(i)}) \cdot (1 - \text{rel}^{\text{nn}, \alpha}(\mathbf{y}^{(i-1)}))$  // convert  $\mathbf{y}$  to a point on the 2-simplex
- 11     $j_1 \leftarrow \min \{C(\mathbf{y}^{(i-1)}), \hat{C}_{\text{nn}}^\alpha(\mathbf{y}^{(i-1)})\}$
- 12     $j_2 \leftarrow \max \{C(\mathbf{y}^{(i-1)}), \hat{C}_{\text{nn}}^\alpha(\mathbf{y}^{(i-1)})\}$
- 13     $\mathbf{c}^{(i)} \leftarrow (c_{j_1}^{(i-1)}, c_{j_2}^{(i-1)}, i+1)$  // obtain the new set of 3 colors
- 14  $j \leftarrow C(\mathbf{y}^{(k)})$
- 15 **return**  $c_j^{(k)}$

---

## 7.2 Hard instances with two dimensions

In this subsection, we present our symmetric construction for  $k = 2$ . Our construction converts a point of the 2-simplex to points on another 2-simplex and uses the color of the converted coordinates to guarantee symmetry. Given a point  $\mathbf{x}$  in the 2-simplex, we first project it onto the base 1-simplex. Suppose the projection gives  $(1 - y_0, y_0)$ . We define a continuous and piecewise linear mapping from the 1-simplex to itself:

$$\tilde{y}_0 = \begin{cases} 0 & \text{if } y_0 \leq 0.1 - \varepsilon^2, \\ 0.5 + 0.5\varepsilon^{-2} \cdot (y_0 - 0.1) & \text{if } y_0 \in 0.1 \pm \varepsilon^2, \\ 1 & \text{if } y_0 \geq 0.1 + \varepsilon^2. \end{cases}$$

Finally, we use  $((1 - x_3) \cdot (1 - \tilde{y}_0), (1 - x_3) \cdot \tilde{y}_0, x_3)$  as the intermediate projection  $\mathbf{y}^{(2)}$  of  $\mathbf{x}$  and define the color of  $C_{\text{sym}}^{(2)}(\mathbf{x})$  as the color of  $\mathbf{y}^{(2)}$  in the base instance  $C$ . The first 5 lines of Algorithm 4 concludes our symmetric construction for  $k = 2$ .

**Symmetry.** For the 2-dimensional instances, the symmetry results from the simple calculation of the colors on the three 1-simplices on the boundary.

**Lemma 58.** Algorithm 4 gives a valid 3-out-of- $k+1$  Approximate Symmetric SPERNER instance for  $k = 2$ .

*Proof.* We will give a complete characterization of the coloring on the three boundaries: for any  $x \in [0, 1]$ ,

$$C_{\text{sym}}^{(2)}(x, 1 - x, 0) = 1 + \mathbb{1}(x < 0.9), \quad (17)$$

$$C_{\text{sym}}^{(2)}(x, 0, 1 - x) = 1 + 2 \cdot \mathbb{1}(x < 0.9), \quad (18)$$

$$C_{\text{sym}}^{(2)}(0, x, 1 - x) = 2 + \mathbb{1}(x < 0.9). \quad (19)$$

This directly implies  $(x, 1 - x, 0) \sim_{C_{\text{sym}}^{(2)}} (x, 0, 1 - x) \sim_{C_{\text{sym}}^{(2)}} (0, x, 1 - x)$  for any  $x \in [0, 1]$ .

For  $(x, 1 - x, 0)$ , the projection is trivially  $(x, 1 - x)$ . Then, we discuss three cases to prove [Eq. \(17\)](#):

- If  $x \in 0.9 \pm \varepsilon^2$ ,  $0.5\varepsilon^{-2} \cdot (y_0 - 0.1) \in \pm 0.5$  and thus  $\tilde{y}_0 = 0.5 + 0.5\varepsilon^{-2} \cdot (0.9 - x)$ . The projection  $\mathbf{y}^{(2)}$  in consideration, is then

$$(0.5 - 0.5\varepsilon^{-2} \cdot (0.9 - x), 0.5 + 0.5\varepsilon^{-2} \cdot (0.9 - x), 0) .$$

If  $x < 0.9$ ,  $y_2^{(2)} > 0.5$ . Otherwise if  $x \geq 0.9$ ,  $y_2^{(2)} \leq 0.5$ . Since  $C(1 - y, y, 0) = 1 + \mathbb{1}(y > 0.5)$  ([Fact 25](#)), we have  $C_{\text{sym}}^{(2)}(x, 1 - x, 0) = 1 + \mathbb{1}(x < 0.9)$  for this subcase.

- If  $x > 0.9 + \varepsilon^2$ ,  $0.5\varepsilon^{-2} \cdot (y_0 - 0.1) < -0.5$  and thus  $\tilde{y}_0 = 0$ . The intermediate projection  $\mathbf{y}^{(2)}$  in consideration, is then  $(1, 0, 0)$ . Since  $C(1, 0, 0) = 1$  ([Fact 26](#)), we have  $C_{\text{sym}}^{(2)}(x, 1 - x, 0) = 1$  in this subcase.
- If  $x < 0.9 - \varepsilon^2$ ,  $0.5\varepsilon^{-2} \cdot (y_0 - 0.1) > 0.5$  and thus  $\tilde{y}_0 = 1$ . The intermediate projection  $\mathbf{y}^{(2)}$  in consideration, is then  $(0, 1, 0)$ . Since  $C(0, 1, 0) = 2$  ([Fact 26](#)), we have  $C_{\text{sym}}^{(2)}(x, 1 - x, 0) = 2$  in this subcase.

For  $(x, 0, 1 - x)$ , the projection is  $(1, 0)$ . Therefore,  $\tilde{y}_0 = 0$  and the intermediate projection  $\mathbf{y}^{(2)} = (x, 0, 1 - x)$ . According to [Fact 26](#),  $C_{\text{sym}}^{(2)}(x, 0, 1 - x) = C(x, 0, 1 - x) = 1 + 2 \cdot \mathbb{1}(x < 0.9)$ , matching [Eq. \(18\)](#).

For  $(0, x, 1 - x)$ , the projection is  $(0, 1)$ . Therefore,  $\tilde{y}_0 = 1$  and the intermediate projection  $\mathbf{y}^{(2)} = (0, x, 1 - x)$ . According to [Fact 26](#),  $C_{\text{sym}}^{(2)}(0, x, 1 - x) = C(0, x, 1 - x) = 2 + \mathbb{1}(x < 0.9)$ , matching [Eq. \(19\)](#).  $\square$

**Hardness.** Note that we embed a PPAD-hard instance  $C$  inside  $C_{\text{sym}}^{(2)}$  with a scaling factor of  $2^{-2n+1} = 2\varepsilon^2$  on the first and the second coordinates. It is then easy to obtain the following PPAD-hardness result for  $C_{\text{sym}}^{(2)}$ .

**Lemma 59.** *It is PPAD-hard to find three points  $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \mathbf{x}^{(3)}$  such that*

- **they are close enough to each other:** for any  $i, j \in [3]$ ,  $\|\mathbf{x}^{(i)} - \mathbf{x}^{(j)}\|_{\infty} \leq 2^{-3n} = \varepsilon^3$ ,
- **they induce a trichromatic triangle:**  $|\{C_{\text{sym}}^{(2)}(\mathbf{x}^{(i)}) : i \in [3]\}| = 3$ .

### 7.3 Hard instances with three or more dimensions

For three or higher dimensions, we use almost the same recursive definition as in [Section 6](#). We repeat it in [Algorithm 4](#) for convenience. In the rest of this subsection, we will show that our construction is symmetric and still PPAD-hard. Here, we use the same way to define the *modified neighboring color*:

$$\hat{C}_{\text{nn}}^{\alpha}(\mathbf{x}) = \begin{cases} C_{\text{nn}}^{\alpha}(\mathbf{x}) & \text{if } d^{\alpha}(\mathbf{x}, \mathbf{nn}^{\alpha}(\mathbf{x})) < 2\varepsilon^2, \\ 2 & \text{if } d^{\alpha}(\mathbf{x}, \mathbf{nn}^{\alpha}(\mathbf{x})) \geq 2\varepsilon^2 \text{ and } C(\mathbf{x}) = 1, \\ 1 & \text{if } d^{\alpha}(\mathbf{x}, \mathbf{nn}^{\alpha}(\mathbf{x})) \geq 2\varepsilon^2 \text{ and } C(\mathbf{x}) \in \{2, 3\}. \end{cases} \quad (20)$$

**Fact 60.** *For any  $\mathbf{x} \in \Delta^2$ , we have  $C(\mathbf{x}) < \hat{C}_{\text{nn}}^{\alpha}(\mathbf{x})$  if  $\text{rel}^{\text{nn}, \alpha}(\mathbf{x}) = 0$ , and  $C(\mathbf{x}) > \hat{C}_{\text{nn}}^{\alpha}(\mathbf{x})$  if  $\text{rel}^{\text{nn}, \alpha}(\mathbf{x}) = 1$ .*

We continue to use  $\mathbf{y}^{(i)}(\mathbf{x}) = (y_1^{(i)}(\mathbf{x}), y_2^{(i)}(\mathbf{x}), y_3^{(i)}(\mathbf{x}))$  to denote the intermediate projections we compute in [Algorithm 4](#) when the input is  $\mathbf{x}$ , and use  $\mathbf{c}^{(i)}(\mathbf{x}) = (c_1^{(i)}(\mathbf{x}), c_2^{(i)}(\mathbf{x}), c_3^{(i)}(\mathbf{x}))$  to denote the intermediate palettes we use in [Algorithm 4](#). In addition, we introduce a new notation  $\tilde{y}_0(\mathbf{x})$  for the converted coordinate we compute on [Line 3](#). Because the subscripts we will use for  $\mathbf{c}^{(i)}$  can be very complicated, we will use

$c_j^{(i)}(\mathbf{x})$  and  $c^{(i)}(\mathbf{x}, j)$  interchangeably for better presentation. For any vector  $\mathbf{y} \in \Delta^2$ , we use  $i^*(\mathbf{y})$  to denote the first non-zero index of  $\mathbf{y}$ , i.e.,

$$i^*(\mathbf{y}) = \begin{cases} 1 & \text{if } y_1 > 0, \\ 2 & \text{if } y_1 = 0 \text{ and } y_2 > 0, \\ 3 & \text{otherwise.} \end{cases}$$

### 7.3.1 Symmetry

To show that  $C^{(k)}$  are valid 3-out-of- $k+1$  Approximate Symmetric SPERNER instances, we consider a stronger symmetry property of our construction for our induction hypothesis. In this stronger symmetry, we consider the following two colors for each point  $\mathbf{x}$ :

1. **the output color:**  $C_{\text{sym}}^{(k)}(\mathbf{x}) = c^{(k)}(\mathbf{x}, C(\mathbf{y}^{(k)}(\mathbf{x})))$  (by definition of [Algorithm 4](#)).
2. **the final neighboring color:** we use  $C_{\text{nn}}^{(k)}(\mathbf{x})$  to denote the intermediate color  $c^{(k)}(\mathbf{x}, \hat{C}_{\text{nn}}^\alpha(\mathbf{y}^{(k)}(\mathbf{x})))$ , which further equals to  $c^{(k)}(\mathbf{x}, C_{\text{nn}}^\alpha(\mathbf{y}^{(k)}(\mathbf{x})))$  when  $\mathbf{y}^{(k)}(\mathbf{x})$  is hot.

The stronger symmetry states that symmetric points  $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}$  on the boundaries are symmetric both in the output color and the final neighboring color. Further, if we try to convert their last intermediate projections,  $\mathbf{y}^{(k)}(\mathbf{x}^{(1)})$  and  $\mathbf{y}^{(k)}(\mathbf{x}^{(2)})$ , we can obtain equivalent converted coordinates. For convenience, we also restate the definition of the symmetry property we are going to establish.

**Definition 12** (symmetric points in a coloring). *We say  $\mathbf{x}$  and  $\mathbf{y}$  are symmetric in a coloring  $C$  if*

- *the number of non-trivial entries in  $\mathbf{x}$  and  $\mathbf{y}$  are the same, i.e.,  $|\mathcal{I}_{>0}(\mathbf{x})| = |\mathcal{I}_{>0}(\mathbf{y})|$ .*
- *the indexing of  $C$  are the same for the arrays of non-trivial indices of  $\mathbf{x}$  and  $\mathbf{y}$ , i.e.,  $\text{index}(\mathcal{I}_{>0}(\mathbf{x}), C(\mathbf{x})) = \text{index}(\mathcal{I}_{>0}(\mathbf{y}), C(\mathbf{y}))$ .*

We use  $\mathbf{x} \sim_C \mathbf{y}$  to denote this symmetry.

**Definition 14** (3-out-of- $k+1$  Approximate Symmetric SPERNER). *3-out-of- $k+1$  Approximate Symmetric SPERNER is a special case of 3-out-of- $k+1$  Approximate SPERNER, where the given circuit  $C$  further satisfies the following property: for any  $\mathbf{x} \in \Delta_n^k$  and any  $i, j \in [k+1]$ ,  $(\mathbf{x}_{1:i-1}, 0, \mathbf{x}_{i:k}) \sim_C (\mathbf{x}_{1:j-1}, 0, \mathbf{x}_{j:k})$ .*

**Lemma 61.** *Consider any  $k \geq 2$ , any two indices  $j_1, j_2 \in [k+1]$  and any  $\mathbf{x}^{(1)}, \mathbf{x}^{(2)} \in \Delta^k$  such that  $x_{j_1}^{(1)} = x_{j_2}^{(2)} = 0$  and  $\mathbf{x}_{-j_1}^{(1)} = \mathbf{x}_{-j_2}^{(2)}$ .  $\mathbf{x}^{(1)}$  and  $\mathbf{x}^{(2)}$  are*

- **symmetric in the output color:**  $\mathbf{x}^{(1)} \sim_{C_{\text{sym}}^{(k)}} \mathbf{x}^{(2)}$ ;
- **their subsequent converted coordinates are equivalent:**  $\text{rel}^{\text{nn}, \alpha}(\mathbf{y}^{(k)}(\mathbf{x}^{(1)})) \sim_{\text{rel}} \text{rel}^{\text{nn}, \alpha}(\mathbf{y}^{(k)}(\mathbf{x}^{(2)}))$ ;  
and
- **symmetric in the final neighboring color:**  $\mathbf{x}^{(1)} \sim_{C_{\text{nn}}^{(k)}} \mathbf{x}^{(2)}$  if  $\mathbf{y}^{(k)}(\mathbf{x}^{(1)}), \mathbf{y}^{(k)}(\mathbf{x}^{(2)})$  are hot.

Note that, in this lemma, the first bullet directly implies the symmetry property we desire for Symmetric SPERNER instances ([Definition 14](#)).

**Corollary 62.** *For any  $k \geq 2$ ,  $C_{\text{sym}}^{(k)}$  constructed by [Algorithm 4](#) is a valid 3-out-of- $k+1$  Approximate Symmetric SPERNER instance.*

Next, we establish this stronger symmetry by induction. The base case is when  $k = 2$ , which we will prove by simple calculations. Later, we will establish this stronger symmetry for  $k = k_0 \geq 3$  under the assumption that we have established it for every  $k < k_0$ . More specifically, we shall discuss three cases on the indices  $j_1, j_2$  of asymmetric zero entries of  $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}$ :

$$(1) j_1, j_2 \leq k;$$

$$(2) j_1 = k \text{ and } j_2 = k + 1;$$

$$(3) j_1 < k \text{ and } j_2 = k + 1.$$

Case (1) can be interpreted as appending a same  $k + 1$ -th coordinate to two vectors  $\mathbf{x}_{-(k+1)}^{(1)}, \mathbf{x}_{-(k+1)}^{(2)}$  that are symmetric in the smaller instance defined on  $\Delta^{k-1}$ . We will use our induction hypothesis to establish this case. Case (2) can be interpreted as appending a same (possibly non-zero) coordinate and a zero coordinate to the same vector  $\mathbf{x}_{1:k-1}^{(1)} = \mathbf{x}_{1:k-1}^{(2)}$  but in different orders for  $\mathbf{x}^{(1)}$  and  $\mathbf{x}^{(2)}$ . We will simulate the algorithm to establish this case. Finally, Case (3) include all the other cases and can be decomposed into a piece of Case (1) and a piece of Case (2), by finding an intermediate vector  $\mathbf{x}^{(3)}$  such that the symmetry between  $\mathbf{x}^{(1)}, \mathbf{x}^{(3)}$  follows Case (1) and the symmetry between  $\mathbf{x}^{(2)}, \mathbf{x}^{(3)}$  follows Case (2). We will simply establish this case by the fact that symmetry is an equivalence relation (Fact 13).

One useful lemma is that the symmetry between points is preserved under the max/min operations on the colorings.

**Lemma 63.** *For any two colorings  $C_1, C_2$  and two vectors  $\mathbf{x}, \mathbf{y}$ , if  $\mathbf{x} \sim_{C_1} \mathbf{y}$  and  $\mathbf{x} \sim_{C_2} \mathbf{y}$ , they are also symmetric in their max-coloring  $\max(C_1, C_2)$  (i.e.,  $\max(C_1, C_2)(\mathbf{x}) = \max\{C_1(\mathbf{x}), C_2(\mathbf{x})\}$  for any  $\mathbf{x}$ ) and min-coloring  $\min(C_1, C_2)$  (resp.,  $\max(C_1, C_2)(\mathbf{x}) = \max\{C_1(\mathbf{x}), C_2(\mathbf{x})\}$  for any  $\mathbf{x}$ ).*

*Proof.* According to Definition 12, the given symmetries both imply  $|\mathcal{I}_{>0}(\mathbf{x})| = |\mathcal{I}_{>0}(\mathbf{y})|$ . For simplicity, we let  $\mathcal{I}_x = \mathcal{I}_{>0}(\mathbf{x})$  and  $\mathcal{I}_y = \mathcal{I}_{>0}(\mathbf{y})$ . In addition,  $\mathbf{x} \sim_{C_1} \mathbf{y}$  implies  $\text{index}(\mathcal{I}_x, C_1(\mathbf{x})) = \text{index}(\mathcal{I}_y, C_1(\mathbf{y}))$ , while  $\mathbf{x} \sim_{C_2} \mathbf{y}$  implies  $\text{index}(\mathcal{I}_x, C_2(\mathbf{x})) = \text{index}(\mathcal{I}_y, C_2(\mathbf{y}))$ . If  $C_1(\mathbf{x}) = C_2(\mathbf{x})$ , we have  $C_1(\mathbf{y}) = C_2(\mathbf{y})$  and the lemma trivially holds. W.l.o.g., we suppose  $C_1(\mathbf{x}) < C_2(\mathbf{x})$  next. Note that, according to Definition 11, for any increasing array with positive integer entries  $\mathbf{a}$ ,  $\text{index}(\mathbf{a}, v)$  is a strictly increasing function for  $v \in \mathbf{a}$ . We have  $\text{index}(\mathcal{I}_x, C_1(\mathbf{x})) < \text{index}(\mathcal{I}_x, C_2(\mathbf{x}))$  and thus  $\text{index}(\mathcal{I}_y, C_1(\mathbf{y})) < \text{index}(\mathcal{I}_y, C_2(\mathbf{y}))$ . The later inequality further implies  $C_1(\mathbf{y}) < C_2(\mathbf{y})$ . Therefore,  $\min(C_1, C_2)(\mathbf{x}) = C_1(\mathbf{x})$  and  $\min(C_1, C_2)(\mathbf{y}) = C_1(\mathbf{y})$ . We have  $\text{index}(\mathcal{I}_x, \min(C_1, C_2)(\mathbf{x})) = \text{index}(\mathcal{I}_y, \min(C_1, C_2)(\mathbf{y}))$ . According to Definition 12, we have  $\mathbf{x} \sim_{\min(C_1, C_2)} \mathbf{y}$ . Similarly, we have  $\max(C_1, C_2)(\mathbf{x}) = C_2(\mathbf{x})$  and  $\max(C_1, C_2)(\mathbf{y}) = C_2(\mathbf{y})$ , and  $\mathbf{x} \sim_{\max(C_1, C_2)} \mathbf{y}$ .  $\square$

**The base case ( $k = 2$ ).** The first bullet for the base case has been established by Lemma 58. Next, we consider any fixed  $z \in [0, 1]$ . We will establish the second and the third bullet for  $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}$  that are in the set  $\{(1 - z, z, 0), (1 - z, 0, z), (0, 1 - z, z)\}$ . According to the first two lines of Algorithm 4,  $\tilde{y}_0(1 - z, z, 0) = (0.5 + 0.5\epsilon^{-2} \cdot (z - 0.1))_{[0,1]}$ ,  $\tilde{y}_0(1 - z, 0, z) = 0$  and  $\tilde{y}_0(0, 1 - z, z) = 1$ . Therefore, we have

$$\begin{aligned} \mathbf{y}^{(2)}(1 - z, z, 0) &= \left( (0.5 - 0.5\epsilon^{-2} \cdot (z - 0.1))_{[0,1]}, (0.5 + 0.5\epsilon^{-2} \cdot (z - 0.1))_{[0,1]}, 0 \right), \\ \mathbf{y}^{(2)}(1 - z, 0, z) &= (1 - z, 0, z), \\ \mathbf{y}^{(2)}(0, 1 - z, z) &= (0, 1 - z, z). \end{aligned}$$

According to Lemmas 51 and 53, we can obtain the second bullet for  $k = 2$ :

$$\begin{aligned} \text{rel}^{nn,\alpha}(\mathbf{y}^{(2)}(1 - z, 0, z)) &\sim_{\text{rel}} \text{rel}^{nn,\alpha}(\mathbf{y}^{(2)}(0, 1 - z, z)) \sim_{\text{rel}} \\ &\text{rel}^{nn,\alpha}(\mathbf{y}^{(2)}(1 - z, z, 0)) = (0.5 + 0.5\epsilon^{-2} \cdot (z - 0.1))_{[0,1]}. \end{aligned}$$

This equivalence on the subsequent converted coordinate implies that  $\mathbf{y}^{(2)}(\mathbf{x}^{(1)}), \mathbf{y}^{(2)}(\mathbf{x}^{(2)})$  are hot if and only if  $z \in 0.1 \pm \epsilon^2$ . Further, according to Definition 12, Lemmas 51 and 53, it is easy to obtain  $(1 - z, z, 0) \sim_{C_{\text{nn}}^{(2)}} (1 - z, 0, z) \sim_{C_{\text{nn}}^{(2)}} (0, 1 - z, z)$ .

**General case 1 ( $k \geq 3$  and  $j_1, j_2 \leq k$ ).** In this subcase, we have  $x_{k+1}^{(1)} = x_{k+1}^{(2)}$ . We can w.l.o.g. assume that  $x_{k+1}^{(1)} < 1$  because otherwise  $\mathbf{x}^{(1)} = \mathbf{x}^{(2)}$  and this stronger symmetry trivially holds. Let  $\hat{\mathbf{x}}^{(1)} = \mathbf{P}(\mathbf{x}^{(1)})$  and

$\hat{\mathbf{x}}^{(2)} = \mathbf{P}(\mathbf{x}^{(2)})$ . According to the definition of projection operation  $\mathbf{P}$  (Definition 29), we have

$$\hat{\mathbf{x}}^{(1)} = \frac{\mathbf{x}_{1:k}^{(1)}}{1 - x_{k+1}^{(1)}},$$

$$\hat{\mathbf{x}}^{(2)} = \frac{\mathbf{x}_{1:k}^{(2)}}{1 - x_{k+1}^{(2)}} = \frac{\mathbf{x}_{1:k}^{(1)}}{1 - x_{k+1}^{(1)}}.$$

Hence, they satisfy the same symmetry as  $\mathbf{x}^{(1)}$  and  $\mathbf{x}^{(2)}$ , i.e., we have  $\hat{x}_{j_1}^{(1)} = \hat{x}_{j_2}^{(2)} = 0$  and  $(\hat{\mathbf{x}}^{(1)})_{-j_1} = (\hat{\mathbf{x}}^{(2)})_{-j_2}$ . An observation is that the first  $k-2$  intermediate projections we use for any input  $\mathbf{x} \in \Delta^k$  are exactly the same as those for its one-step projection  $\mathbf{P}(\mathbf{x}) \in \Delta^{k-1}$  (in a different run of Algorithm 4 on a  $k-1$ -dimensional simplex).

**Observation 64.** For any  $k \geq 3$ , any  $\mathbf{x} \in \Delta^k$  and any  $i \in [k-1]$ , we have  $\mathbf{y}^{(i)}(\mathbf{x}) = \mathbf{y}^{(i)}(\mathbf{P}(\mathbf{x}))$ .

Using this observation and the second bullet of our induction hypothesis for  $k-1$ , we can easily obtain the equivalence between the converted coordinates of the  $(k-1)$ -th intermediate projections  $\mathbf{y}^{(k-1)}(\cdot)$  of  $\mathbf{x}^{(1)}$  and  $\mathbf{x}^{(2)}$ .

$$\text{rel}^{nn,\alpha}(\mathbf{y}^{(k-1)}(\mathbf{x}^{(1)})) = \text{rel}^{nn,\alpha}(\mathbf{y}^{(k-1)}(\hat{\mathbf{x}}^{(1)})) \sim_{\text{rel}} \text{rel}^{nn,\alpha}(\mathbf{y}^{(k-1)}(\hat{\mathbf{x}}^{(2)})) = \text{rel}^{nn,\alpha}(\mathbf{y}^{(k-1)}(\mathbf{x}^{(2)})). \quad (21)$$

Since Line 8 gives  $y_3^{(k)}(\mathbf{x}^{(1)}) = x_{k+1}^{(1)}$  and  $y_3^{(k)}(\mathbf{x}^{(2)}) = x_{k+1}^{(2)}$ , we clearly have  $y_3^{(k)}(\mathbf{x}^{(1)}) = y_3^{(k)}(\mathbf{x}^{(2)})$ . For simplicity, we let  $z = y_3^{(k)}(\mathbf{x}^{(1)})$  next. According to Lines 9 and 10 of Algorithm 4 and because of Eq. (21), we also have either

- $y_2^{(k)}(\mathbf{x}^{(1)}) = y_2^{(k)}(\mathbf{x}^{(2)}) \in (0, 1-z)$ , which implies  $\mathbf{y}^{(k)}(\mathbf{x}^{(1)}) = \mathbf{y}^{(k)}(\mathbf{x}^{(2)})$ , or
- $y_2^{(k)}(\mathbf{x}^{(1)}), y_2^{(k)}(\mathbf{x}^{(2)}) \in \{0, 1-z\}$ .

According to Lemma 53, we have  $\text{rel}^{nn,\alpha}(\mathbf{y}^{(k)}(\mathbf{x}^{(1)})) \sim_{\text{rel}} \text{rel}^{nn,\alpha}(\mathbf{y}^{(k)}(\mathbf{x}^{(2)}))$ , the second bullet of Lemma 61. Next, we prove the first bullet and the third bullet by further discussing these two subcases. Recall the following definitions:

$$C_{\text{sym}}^{(k)}(\mathbf{x}) := c^{(k)}(\mathbf{x}, C(\mathbf{y}^{(k)}(\mathbf{x}))) ,$$

$$C_{\text{nn}}^{(k)}(\mathbf{x}) := c^{(k)}(\mathbf{x}, C_{\text{nn}}^\alpha(\mathbf{y}^{(k)}(\mathbf{x}))) , \quad \text{when } \mathbf{y}^{(k)}(\mathbf{x}) \text{ is hot} .$$

First, we consider the subcase in the earlier first bullet. That is, we suppose that  $y_2^{(k)}(\mathbf{x}^{(1)}) = y_2^{(k)}(\mathbf{x}^{(2)}) \in (0, 1-z)$ , which implies  $\mathbf{y}^{(k)}(\mathbf{x}^{(1)}) = \mathbf{y}^{(k)}(\mathbf{x}^{(2)})$ . According to Line 9, we have

$$\text{rel}^{nn,\alpha}(\mathbf{y}^{(k-1)}(\mathbf{x}^{(1)})), \text{rel}^{nn,\alpha}(\mathbf{y}^{(k-1)}(\mathbf{x}^{(2)})) \in (0, 1) ,$$

and thus the  $(k-1)$ -th intermediate projections  $\mathbf{y}^{(k-1)}(\mathbf{x}^{(1)})$  and  $\mathbf{y}^{(k-1)}(\mathbf{x}^{(2)})$  are hot. Recall that our induction hypothesis gives that  $\hat{\mathbf{x}}^{(1)} \sim_{C_{\text{sym}}^{(k-1)}} \hat{\mathbf{x}}^{(2)}$  and  $\hat{\mathbf{x}}^{(1)} \sim_{C_{\text{nn}}^{(k-1)}} \hat{\mathbf{x}}^{(2)}$ . According to Line 13, the first colors in the intermediate color palette  $c^{(k)}$  of any  $\mathbf{x}$  form the set  $\{C_{\text{sym}}^{(k-1)}(\mathbf{P}(\mathbf{x})), C_{\text{nn}}^{(k-1)}(\mathbf{P}(\mathbf{x}))\}$ , in an order where  $c_1^{(k)}(\mathbf{x}) < c_2^{(k)}(\mathbf{x})$ . Because min/max operations on the two colorings preserve their symmetry (Lemma 63),  $\mathbf{x}^{(1)}$  and  $\mathbf{x}^{(2)}$  are symmetric in the intermediate colors  $c_1^{(k)}(\cdot)$  and  $c_2^{(k)}(\cdot)$ , i.e.,

$$\mathbf{x}^{(1)} \sim_{c_1^{(k)}(\cdot)} \mathbf{x}^{(2)} , \quad \text{and} \quad \mathbf{x}^{(1)} \sim_{c_2^{(k)}(\cdot)} \mathbf{x}^{(2)} .$$

Note that  $c_3^{(k)}(\mathbf{x}^{(1)}) = k+1 = c_3^{(k)}(\mathbf{x}^{(2)})$ . Then, we clearly have the first and the third bullet of Lemma 61,  $\mathbf{x}^{(1)} \sim_{C_{\text{sym}}^{(k)}} \mathbf{x}^{(2)}$  and  $\mathbf{x}^{(1)} \sim_{C_{\text{nn}}^{(k)}} \mathbf{x}^{(2)}$  (when  $\mathbf{y}^{(k)}(\mathbf{x}^{(1)})$  and  $\mathbf{y}^{(k)}(\mathbf{x}^{(2)})$  are hot), because we trivially have  $C(\mathbf{y}^{(k)}(\mathbf{x}^{(1)})) = C(\mathbf{y}^{(k)}(\mathbf{x}^{(2)}))$  and  $C_{\text{nn}}^\alpha(\mathbf{y}^{(k)}(\mathbf{x}^{(1)})) = C_{\text{nn}}^\alpha(\mathbf{y}^{(k)}(\mathbf{x}^{(2)}))$  by  $\mathbf{y}^{(k)}(\mathbf{x}^{(1)}) = \mathbf{y}^{(k)}(\mathbf{x}^{(2)})$ .

On the other hand, we consider the subcase in the earlier second bullet. That is, we suppose that  $y_2^{(k)}(\mathbf{x}^{(1)}), y_2^{(k)}(\mathbf{x}^{(2)}) \in \{0, 1 - z\}$  (i.e., we have  $\mathbf{y}^{(k)}(\mathbf{x}^{(1)}), \mathbf{y}^{(k)}(\mathbf{x}^{(2)}) \in \{(1 - z, 0, z), (0, 1 - z, z)\}$ ). Our induction hypothesis only gives us  $\hat{\mathbf{x}}^{(1)} \sim_{C_{\text{sym}}^{(k-1)}} \hat{\mathbf{x}}^{(2)}$  here. According to [Line 9](#), we have

$$\text{rel}^{nn,\alpha}(\mathbf{y}^{(k-1)}(\mathbf{x}^{(1)}), \text{rel}^{nn,\alpha}(\mathbf{y}^{(k-1)}(\mathbf{x}^{(2)})) \in \{0, 1\},$$

and thus the  $(k - 1)$ -th intermediate projections  $\mathbf{y}^{(k-1)}(\mathbf{x}^{(1)})$  and  $\mathbf{y}^{(k-1)}(\mathbf{x}^{(2)})$  are not hot. Note that we have  $C_{\text{sym}}^{(k-1)}(\hat{\mathbf{x}}^{(1)}) = c^{(k-1)}(\mathbf{x}^{(1)}, C(\mathbf{y}^{(k-1)}(\mathbf{x}^{(1)}))) < c^{(k-1)}(\mathbf{x}^{(1)}, \hat{C}_{\text{nn}}^\alpha(\mathbf{y}^{(k-1)}(\mathbf{x}^{(1)})))$  if and only if  $y_2^{(k)}(\mathbf{x}^{(1)}) = 0$  (similarly, for  $\mathbf{x}^{(2)}$ ). Therefore, according to [Line 13](#) of [Algorithm 4](#), we have

$$c^{(k)}(\mathbf{x}^{(1)}, i^*(\mathbf{y}^{(k)}(\mathbf{x}^{(1)}))) = C_{\text{sym}}^{(k-1)}(\hat{\mathbf{x}}^{(1)}) \quad \text{and} \quad c^{(k)}(\mathbf{x}^{(2)}, i^*(\mathbf{y}^{(k)}(\mathbf{x}^{(2)}))) = C_{\text{sym}}^{(k-1)}(\hat{\mathbf{x}}^{(2)}).$$

Note we have  $C(1 - z, 0, z) = 1, C(0, 1 - z, z) = 2$  (i.e.,  $C(1 - z, 0, z) = i^*(1 - z, 0, z)$  and  $C(0, 1 - z, z) = i^*(0, 1 - z, z)$ ), or  $C(1 - z, 0, z) = C(0, 1 - z, z) = 3$  for any fixed  $z \in [0, 1]$  ([Fact 26](#)), and the same thing holds if we replace  $C$  by  $C_{\text{nn}}^\alpha$  ([Lemma 53](#)) and we have  $\text{rel}^{nn,\alpha}(1 - z, 0, z) \in (0, 1)$ . Because  $C_{\text{sym}}^{(k)}(\mathbf{x})$  is defined as  $c^{(k)}(\mathbf{x}, C(\mathbf{y}^{(k)}(\mathbf{x})))$  and because of our induction hypothesis that  $\hat{\mathbf{x}}^{(1)} \sim_{C_{\text{sym}}^{(k-1)}} \hat{\mathbf{x}}^{(2)}$ , we have  $\mathbf{x}^{(1)} \sim_{C_{\text{sym}}^{(k)}} \mathbf{x}^{(2)}$  and  $\mathbf{x}^{(1)} \sim_{C_{\text{nn}}^{(k)}} \mathbf{x}^{(2)}$  if  $\mathbf{y}^{(k)}(\mathbf{x}^{(1)})$  is hot (i.e.,  $\text{rel}^{nn,\alpha}(\mathbf{y}^{(k)}(\mathbf{x}^{(1)})) \in (0, 1)$ ).

**General case 2** ( $k \geq 3, j_1 = k$  and  $j_2 = k + 1$ ). In this case, we have  $x_{k+1}^{(1)} = x_k^{(2)}$  and  $x_k^{(1)} = x_{k+1}^{(2)} = 0$ . Let  $\hat{\mathbf{x}}^{(1)} = \mathbf{P}^{(2)}(\mathbf{x}^{(1)})$  and  $\hat{\mathbf{x}}^{(2)} = \mathbf{P}^{(2)}(\mathbf{x}^{(2)})$ . According to the definition of the projection step  $\mathbf{P}$  ([Definition 29](#)), we have

$$\begin{aligned} \hat{\mathbf{x}}^{(1)} &= \frac{\mathbf{x}_{1:k-1}^{(1)}}{1 - x_{k+1}^{(1)}}, \\ \hat{\mathbf{x}}^{(2)} &= \frac{\mathbf{x}_{1:k-1}^{(2)}}{1 - x_k^{(2)}} = \frac{\mathbf{x}_{1:k-1}^{(2)}}{1 - x_{k+1}^{(1)}}. \end{aligned}$$

Because  $\mathbf{x}_{-(k+1)}^{(1)} = \mathbf{x}_{-k}^{(2)}$ , which implies  $\mathbf{x}_{1:k-1}^{(1)} = \mathbf{x}_{1:k-1}^{(2)}$ , we have  $\hat{\mathbf{x}}^{(1)} = \hat{\mathbf{x}}^{(2)}$ .

Note that  $k \geq 3$ . We have  $\hat{\mathbf{x}}^{(1)}, \hat{\mathbf{x}}^{(2)} \in \Delta^{k-2}$ . Therefore, we have  $\mathbf{P}^{(k-2)}(\mathbf{x}^{(1)}) = \mathbf{P}^{(k-2)}(\mathbf{x}^{(2)})$  and further  $\tilde{y}_0(\mathbf{x}^{(1)}) = \tilde{y}_0(\mathbf{x}^{(2)})$ . In addition, we trivially have  $\mathbf{c}^{(2)}(\mathbf{x}^{(1)}) = (1, 2, 3) = \mathbf{c}^{(2)}(\mathbf{x}^{(2)})$ . Similarly, according to [Observation 64](#), we can prove that  $\mathbf{y}^{(k-2)}(\mathbf{x}^{(1)}) = \mathbf{y}^{(k-2)}(\mathbf{x}^{(2)})$  and  $\mathbf{c}^{(k-1)}(\mathbf{x}^{(1)}) = \mathbf{c}^{(k-1)}(\mathbf{x}^{(2)})$  if  $k \geq 4$  (because [Algorithm 4](#) computes  $\mathbf{c}^{(k-1)}(\mathbf{x})$  only based on  $\mathbf{y}^{(k-2)}(\mathbf{x})$ ).

For simplicity, we let  $\tilde{y} = \text{rel}^{nn,\alpha}(\mathbf{y}^{(k-2)}(\mathbf{x}^{(1)}))$  if  $k \geq 4$  and let  $\tilde{y} = \tilde{y}_0(\mathbf{x}^{(1)})$  if  $k = 3$ . According to our previous discussions, we have  $\tilde{y} = \text{rel}^{nn,\alpha}(\mathbf{y}^{(k-2)}(\mathbf{x}^{(2)}))$  if  $k \geq 4$  and  $\tilde{y} = \tilde{y}_0(\mathbf{x}^{(2)})$  if  $k = 3$ . Further we let  $z := x_{k+1}^{(1)}$  and  $(c_1, c_2, c_3) := \mathbf{c}^{(k-1)}(\mathbf{x}^{(1)}) \in [k]^3$ , which clearly gives  $c_3 = k$  according to [Line 13](#) of [Algorithm 4](#). Again, according to our previous discussion, we have  $z = x_k^{(2)}$  and  $\mathbf{c}^{(k-1)}(\mathbf{x}^{(2)}) = (c_1, c_2, c_3)$ . Also, recall that  $x_k^{(1)} = x_{k+1}^{(2)} = 0$ . Finally, we use  $\mathbf{y}^* := ((1 - z) \cdot (1 - \tilde{y}), (1 - z) \cdot \tilde{y}, z)$  to denote a key vector in our following proof, and use  $\tilde{y}^* := \text{rel}^{nn,\alpha}(\mathbf{y}^*)$  to denote its converted coordinate.

First, we prove the second bullet of [Lemma 61](#),  $\text{rel}^{nn,\alpha}(\mathbf{y}^{(k)}(\mathbf{x}^{(1)})) = \text{rel}^{nn,\alpha}(\mathbf{y}^{(k)}(\mathbf{x}^{(2)}))$ , by the following simple simulations for [Algorithm 4](#), where we prove that  $\text{rel}^{nn,\alpha}(\mathbf{y}^{(k)}(\mathbf{x}^{(1)})) = \text{rel}^{nn,\alpha}(\mathbf{y}^{(k)}(\mathbf{x}^{(2)})) = \tilde{y}^*$ .

- For  $\mathbf{x}^{(1)}$ , we have  $\mathbf{y}^{(k-1)}(\mathbf{x}^{(1)}) = (1 - \tilde{y}, \tilde{y}, 0)$ . According to [Lemma 51](#),  $\text{rel}^{nn,\alpha}(\mathbf{y}^{(k-1)}(\mathbf{x}^{(1)})) = \tilde{y}$ . Therefore, we have  $\mathbf{y}^{(k)}(\mathbf{x}^{(1)}) = ((1 - z) \cdot (1 - \tilde{y}), (1 - z) \cdot \tilde{y}, z) = \mathbf{y}^*$  and thus  $\text{rel}^{nn,\alpha}(\mathbf{y}^{(k)}(\mathbf{x}^{(1)})) = \tilde{y}^*$ .
- For  $\mathbf{x}^{(2)}$ , we have  $\mathbf{y}^{(k-1)}(\mathbf{x}^{(2)}) = ((1 - z) \cdot (1 - \tilde{y}), (1 - z) \cdot \tilde{y}, z) = \mathbf{y}^*$ . According to our definition of  $\tilde{y}^*$ ,  $\text{rel}^{nn,\alpha}(\mathbf{y}^{(k)}(\mathbf{x}^{(2)})) = \tilde{y}^*$  and thus  $\mathbf{y}^{(k)}(\mathbf{x}^{(2)}) = (1 - \tilde{y}^*, \tilde{y}^*, 0)$ . According to [Lemma 51](#), we have  $\text{rel}^{nn,\alpha}(\mathbf{y}^{(k)}(\mathbf{x}^{(2)})) = \tilde{y}^*$ .

Next, we prove the first and the third bullet of [Lemma 61](#),  $\mathbf{x}^{(1)} \sim_{C_{\text{sym}}^{(k)}} \mathbf{x}^{(2)}$  and  $\mathbf{x}^{(1)} \sim_{C_{\text{nn}}^{(k)}} \mathbf{x}^{(2)}$ . Our proof is also based on simulations for [Algorithm 4](#), where we give characterizations of  $C_{\text{sym}}^{(k)}(\mathbf{x}^{(1)})$ ,  $C_{\text{nn}}^{(k)}(\mathbf{x}^{(1)})$  and  $C_{\text{sym}}^{(k)}(\mathbf{x}^{(2)})$ ,  $C_{\text{nn}}^{(k)}(\mathbf{x}^{(2)})$  by [Eqs. \(22\)](#) and [\(23\)](#), respectively.

- For  $\mathbf{x}^{(1)}$ , according to our previous calculations,  $\mathbf{y}^{(k-1)}(\mathbf{x}^{(1)}) = (1 - \tilde{y}, \tilde{y}, 0)$  and  $\mathbf{y}^{(k)}(\mathbf{x}^{(1)}) = \mathbf{y}^*$ . Because of [Fact 25](#), [Lemma 51](#), and [Eq. \(20\)](#),  $\mathbf{y}^{(k-1)}(\mathbf{x}^{(1)})$  is either hot or warm, and we have  $C(\mathbf{y}^{(k-1)}(\mathbf{x}^{(1)}))$ ,  $\hat{C}_{\text{nn}}^\alpha(\mathbf{y}^{(k-1)}(\mathbf{x}^{(1)})) \in \{1, 2\}$ . Therefore,  $\mathbf{c}^{(k)}(\mathbf{x}^{(1)}) = (c_1, c_2, k + 1)$ . The colors  $C_{\text{sym}}^{(k)}$  and  $C_{\text{nn}}^{(k)}$  of  $\mathbf{x}^{(1)}$  are then characterized as follows:

$$C_{\text{sym}}^{(k)}(\mathbf{x}^{(1)}) = \begin{cases} c_1 & \text{if } C(\mathbf{y}^*) = 1, \\ c_2 & \text{if } C(\mathbf{y}^*) = 2, \\ k + 1 & \text{if } C(\mathbf{y}^*) = 3; \end{cases} \quad \text{and} \quad C_{\text{nn}}^{(k)}(\mathbf{x}^{(1)}) = \begin{cases} c_1 & \text{if } \hat{C}_{\text{nn}}^\alpha(\mathbf{y}^*) = 1, \\ c_2 & \text{if } \hat{C}_{\text{nn}}^\alpha(\mathbf{y}^*) = 2, \\ k + 1 & \text{if } \hat{C}_{\text{nn}}^\alpha(\mathbf{y}^*) = 3. \end{cases} \quad (22)$$

- For  $\mathbf{x}^{(2)}$ , according to our previous calculations,  $\mathbf{y}^{(k-1)}(\mathbf{x}^{(2)}) = \mathbf{y}^*$  and  $\mathbf{y}^{(k)}(\mathbf{x}^{(2)}) = (1 - \tilde{y}^*, \tilde{y}^*, 0)$ . We have  $c_1^{(k)}(\mathbf{x}^{(2)}), c_2^{(k)}(\mathbf{x}^{(2)}) \in \{c_i : i \in \{C(\mathbf{y}^*), \hat{C}_{\text{nn}}^\alpha(\mathbf{y}^*)\}\}$ . Because of [Fact 25](#), [Lemma 51](#), and [Eq. \(20\)](#),  $(\tilde{y}^*, 1 - \tilde{y}^*, 0)$  is either hot or warm, and we have  $C(1 - \tilde{y}^*, \tilde{y}^*, 0) \neq \hat{C}_{\text{nn}}^\alpha(1 - \tilde{y}^*, \tilde{y}^*, 0) \in \{1, 2\}$ . Also, note that we have  $C(1 - \tilde{y}^*, \tilde{y}^*, 0) = 1$  if and only if  $\tilde{y}^* \leq 0.5$  if and only if  $C(\mathbf{y}^*) < \hat{C}_{\text{nn}}^\alpha(\mathbf{y}^*)$  ([Eq. \(14\)](#) and [Fact 41](#)) if and only if  $c_1^{(k)}(\mathbf{x}^{(2)}) = c_\ell$  for  $\ell = C(\mathbf{y}^*)$ . Because  $C_{\text{sym}}^{(k)}(\mathbf{x}^{(2)}) = c_\ell^{(k)}(\mathbf{x}^{(2)})$  for  $\ell = C(1 - \tilde{y}^*, \tilde{y}^*, 0)$  and  $C_{\text{nn}}^{(k)}(\mathbf{x}^{(2)}) = c_{\ell'}^{(k)}(\mathbf{x}^{(2)})$  for  $\ell' = \hat{C}_{\text{nn}}^\alpha(1 - \tilde{y}^*, \tilde{y}^*, 0)$ , we have  $C_{\text{sym}}^{(k)}(\mathbf{x}^{(2)}) = c_\ell$  for  $\ell = C(\mathbf{y}^*)$  and  $C_{\text{nn}}^{(k)}(\mathbf{x}^{(2)}) = c_{\ell'}$  for  $\ell' = \hat{C}_{\text{nn}}^\alpha(\mathbf{y}^*)$ . Note that  $c_3 = k$ . We can characterize the colors  $C_{\text{sym}}^{(k)}$  and  $C_{\text{nn}}^{(k)}$  of  $\mathbf{x}^{(2)}$  as follows:

$$C_{\text{sym}}^{(k)}(\mathbf{x}^{(2)}) = \begin{cases} c_1 & \text{if } C(\mathbf{y}^*) = 1, \\ c_2 & \text{if } C(\mathbf{y}^*) = 2, \\ k & \text{if } C(\mathbf{y}^*) = 3; \end{cases} \quad \text{and} \quad C_{\text{nn}}^{(k)}(\mathbf{x}^{(2)}) = \begin{cases} c_1 & \text{if } \hat{C}_{\text{nn}}^\alpha(\mathbf{y}^*) = 1, \\ c_2 & \text{if } \hat{C}_{\text{nn}}^\alpha(\mathbf{y}^*) = 2, \\ k & \text{if } \hat{C}_{\text{nn}}^\alpha(\mathbf{y}^*) = 3. \end{cases} \quad (23)$$

**General case 3** ( $k \geq 3, j_1 < k$  and  $j_2 = k + 1$ ). This case can be easily derived by the equivalence for the earlier two cases and the fact the symmetry in any coloring is an equivalence relation ([Fact 13](#)). That is, letting  $\mathbf{x} = \mathbf{x}_{1:k}^{(2)}$  and  $\mathbf{x}^{(3)} = (\mathbf{x}_{1:k-1}, 0, x_k)$ , the previous two general cases have already given us

$$\begin{aligned} \mathbf{x}^{(1)} &\sim_{C_{\text{sym}}^{(k)}} \mathbf{x}^{(3)} \sim_{C_{\text{sym}}^{(k)}} \mathbf{x}^{(2)}, \\ \text{rel}^{\text{nn}, \alpha}(\mathbf{y}^{(k)}(\mathbf{x}^{(1)})) &\sim_{\text{rel}} \text{rel}^{\text{nn}, \alpha}(\mathbf{y}^{(k)}(\mathbf{x}^{(3)})) \sim_{\text{rel}} \text{rel}^{\text{nn}, \alpha}(\mathbf{y}^{(k)}(\mathbf{x}^{(2)})), \\ \mathbf{x}^{(1)} &\sim_{C_{\text{nn}}^{(k)}} \mathbf{x}^{(3)} \sim_{C_{\text{nn}}^{(k)}} \mathbf{x}^{(2)} \quad \text{if } \text{rel}^{\text{nn}, \alpha}(\mathbf{y}^{(k)}(\mathbf{x}^{(1)})) \in (0, 1). \end{aligned}$$

### 7.3.2 Hardness

It suffices to show PPAD-hardness of the 3-out-of- $k+1$  Approximate Symmetric SPERNER problem by showing that we can always recover a solution for the 2D-SPERNER instance  $C$  in polynomial time.

**Lemma 65.** *Given oracle access to a 2D-SPERNER instance  $C$  and a tuple of points  $(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \mathbf{x}^{(3)})$  which satisfy  $\|\mathbf{x}^{(i)} - \mathbf{x}^{(j)}\|_\infty \leq 2^{-4kn}$  for any  $i, j \in [3]$ , and which are trichromatic in the 3-out-of- $k+1$  Approximate Symmetric SPERNER instance  $C_{\text{sym}}^{(k)}$  constructed by [Algorithm 4](#), then there is a polynomial-time algorithm that finds a tuple of points  $(\hat{\mathbf{x}}^{(1)}, \hat{\mathbf{x}}^{(2)}, \hat{\mathbf{x}}^{(3)})$  which satisfy  $\|\hat{\mathbf{x}}^{(i)} - \hat{\mathbf{x}}^{(j)}\|_\infty \leq 2^{-n}$  for any  $i, j \in [3]$  and which are trichromatic in the 2D-SPERNER instance  $C$ .*



---

**Algorithm 5:** Recover a 2D-SPERNER solution from 3-out-of- $k + 1$  Approximate Symmetric SPERNER solutions

---

**Input :** vectors  $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \mathbf{x}^{(3)} \in \Delta^k$   
**Output :** vectors  $\hat{\mathbf{x}}^{(1)}, \hat{\mathbf{x}}^{(2)}, \hat{\mathbf{x}}^{(3)} \in \Delta^2$

```

1 for  $i \in \{2, 3, \dots, k - 1\}$  do
2   for  $j \in [3]$  do
3     if  $C$  has 3 different colors in  $\mathcal{N}(\mathbf{y}^{(i)}(\mathbf{x}^{(j)}))$  then
4        $\hat{\mathbf{x}}^{(1)}, \hat{\mathbf{x}}^{(2)}, \hat{\mathbf{x}}^{(3)} \leftarrow$  any 3 points colored differently by  $C$  in  $\mathcal{N}(\mathbf{y}^{(i)}(\mathbf{x}^{(j)}))$  // the
5         definition of  $\mathbf{y}^{(i)}(\mathbf{x}^{(j)})$  follows Algorithm 4
6       return  $\hat{\mathbf{x}}^{(1)}, \hat{\mathbf{x}}^{(2)}, \hat{\mathbf{x}}^{(3)}$  // trichromatic triangle found while simulation
6 return  $\mathbf{y}^{(k)}(\mathbf{x}^{(1)}), \mathbf{y}^{(k)}(\mathbf{x}^{(2)}), \mathbf{y}^{(k)}(\mathbf{x}^{(3)})$  // trichromatic triangle found when lifting  $C^{(k-1)}$  to  $C^{(k)}$ 

```

---

The proof for the polynomial-time recovery algorithm (Algorithm 5) is almost the same as that in the warm-up Section 6, as we have established all the analogs of the key lemmas used in the warm-up Section 6. The only differences are

1. We are showing hardness for side-length  $2^{-4kn}$  instead of  $2^{-3kn}$  (caused by Lemma 54 vs. Lemma 37).
2. Proving the Lipschitzness for  $\mathbf{y}^{(2)}(\mathbf{x})$  (as  $\mathbf{y}^{(2)}$  on Line 5 of Algorithm 4 is more sophisticated).
3. Resolving with the issues caused by the fact that  $d^\alpha(\cdot, \cdot)$  is no longer a metric. As we only use the fact  $d(\cdot, \cdot)$  is a metric when we argue in Lemma 45 that points close to a hot point is hot or warm, we prove this argument in Lemma 87 to resolve the issue.

We defer the technical proofs to Appendix A.4.

## 7.4 Query Complexity

Finally, in this subsection, we will establish the query complexity of the continuous version of the 3-out-of- $k + 1$  Approximate Symmetric SPERNER problem, which is  $2^{\Omega(n)}/\text{poly}(n, k)$  (Theorem 47). Our proof will be a careful step-by-step examination on our reduction. Recall that our reduction consists of the following steps:

1. the reduction from the 2D-rectSPERNER problem (instance  $C_{\text{rect}}$ ) to the continuous (triangular) version of the 2D-SPERNER problem (the base instance  $C$ ),
2. the reduction from the continuous version of the 2D-SPERNER problem (the base instance  $C$ ) to the continuous version of the 3-out-of- $k + 1$  Approximate Symmetric SPERNER problem (instance  $C_{\text{sym}}^{(k)}$ ).

**Step 1: reduction from  $C_{\text{rect}}$  to  $C$ .** Recall that our construction of the base instance  $C$  is transforming each point of  $C_{\text{rect}}$  to a  $1.6\epsilon \times 1.6\epsilon$  square and putting it in the corresponding position in the core region of the triangle. To implement the oracle  $C$  for a point inside the core region, we only need to query its corresponding point in  $C_{\text{rect}}$ . For points outside the core region, we can output according to Algorithm 1, without any query to  $C_{\text{rect}}$ . Then, when we receive a small trichromatic triangle in  $C$ , which is always inside the core region according to Fact 28, we can compute the corresponding points of the small triangle in  $C_{\text{rect}}$  to obtain a solution of  $C_{\text{rect}}$  (again, without any query to  $C_{\text{rect}}$ ).

Because the query complexity of  $C_{\text{rect}}$  is  $2^{\Omega(n)}$  according to Theorem 18, we can easily obtain a lower bound of  $2^{\Omega(n)}$  for the query complexity of the base instance  $C$ .

**Lemma 66.** *Suppose  $C$  is constructed by [Algorithm 1](#) using a black-box of  $C_{\text{rect}}$  with side-length of  $2^{-(n-3)}$ . Then, it requires a query complexity of  $2^{\Omega(n)}$  (using oracle  $C$ ) to find a trichromatic region in  $C$ .*

**Step 2: reduction from  $C$  to  $C_{\text{sym}}^{(k)}$ .** Recall our construction [Algorithm 4](#). We use a total number of  $O(k)$  queries to the following functions that require oracle access to  $C$ :

- $C(\mathbf{x})$ : the coloring of the base 2D-SPERNER instance;
- $\hat{C}_{\text{nn}}^\alpha(\mathbf{x})$ : the modified neighboring color; and
- $\text{rel}^{\text{nn},\alpha}(\mathbf{x})$ : the coordinate converter.

Recall that  $\hat{C}_{\text{nn}}^\alpha(\mathbf{x})$  is defined as  $C_{\text{nn}}^\alpha(\mathbf{x})$  when  $\mathbf{x}$  is not cold, and defined according to  $C(\mathbf{x})$  otherwise (see [Eq. \(20\)](#)). Recall that [Lemma 52](#), where we prove that  $\text{rel}^{\text{nn},\alpha}(\mathbf{x})$  and  $C^\alpha(\mathbf{x})$  (when  $\mathbf{x}$  is hot or warm) can be computed in polynomial-time with oracle access to  $C$ . This implies that the last two functions, each  $\text{rel}^{\text{nn},\alpha}(\mathbf{x})$  and  $\hat{C}_{\text{nn}}^\alpha(\mathbf{x})$ , can be computed with a  $\text{poly}(n)$  queries to  $C$ . Therefore, we can construct the instances  $C_{\text{sym}}^{(k)}$  with a total number of  $k \cdot \text{poly}(n)$  queries to  $C$ .

On the other hand, in the recovery algorithm [Algorithm 5](#), we have oracle access to  $C$  for the following process:

- simulating [Algorithm 4](#) for the given solution  $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \mathbf{x}^{(3)}$  for  $C_{\text{sym}}^{(k)}$ ; and
- computing  $\mathcal{N}(\mathbf{y}^{(i)}(\mathbf{x}^{(j)}))$  for each  $i \in \{2, 3, \dots, k-1\}$  and  $j \in [3]$ .

According to earlier discussions, the simulation part can be done in a total number of  $\text{poly}(n, k)$  queries to  $C$ . Recall the definition of the neighborhood ([Definition 36](#)), where the neighbourhood of each point is an  $\varepsilon \times \varepsilon$  square. Because  $C$  is constructed by transforming each point in  $C_{\text{rect}}$  to a  $1.6\varepsilon \times 1.6\varepsilon$  square, we only need to look at at most 8 points in  $C_{\text{rect}}$  to characterize each  $\mathcal{N}(\cdot)$ . Note that, according to [Algorithm 1](#), each  $C_{\text{rect}}(\cdot)$  can be computed by 1 query to  $C$ . Hence, each step in [Algorithm 5](#) using  $\mathcal{N}(\mathbf{y}^{(i)}(\mathbf{x}^{(j)}))$  can be done with a constant number of queries to  $C$ . In total, we can recover a solution for the base instance  $C$  from a solution to  $C_{\text{sym}}^{(k)}$  using a total number of  $\text{poly}(n, k)$  queries to  $C$ .

Suppose the query complexity of finding a trichromatic region in  $C_{\text{sym}}^{(k)}$  is  $f(n, k)$ . According to our previous discussion, we can use our construction algorithm [Algorithm 4](#) and recovery algorithm [Algorithm 5](#) to find a trichromatic region in  $C$  in a total number of  $\text{poly}(n, k) \cdot f(n, k) + \text{poly}(n, k)$  queries to  $C$ . Because finding a trichromatic region in  $C$  requires a query complexity of  $2^{\Omega(n)}$  ([Lemma 66](#)), we have

$$\text{poly}(n, k) \cdot f(n, k) + \text{poly}(n, k) \geq 2^{\Omega(n)} \quad \Rightarrow \quad f(n, k) \geq 2^{\Omega(n)} / \text{poly}(n, k).$$

All of the above discussions can be concluded by the following lemma, which directly implies the query complexity lower-bound in [Theorem 47](#) together with [Lemma 66](#).

**Lemma 67.** *Suppose  $C_{\text{sym}}^{(k)}$  is constructed by [Algorithm 4](#) using a black-box of  $C$  in [Lemma 66](#). Then, it requires a query complexity of  $2^{\Omega(n)} / \text{poly}(n, k)$  (using oracle  $C_{\text{sym}}^{(k)}$ ) to find a trichromatic region in  $C_{\text{sym}}^{(k)}$ .*

## 8 Applications: Envy-Free Cake-Cutting

In this section, we use the hardness result for 3-out-of- $k+1$  Approximate Symmetric SPERNER to obtain a hardness result for the cake-cutting problem. In [Subsection 8.1](#), we focus on the relaxation allowing some (or even most) agents to envy others. Subsequently, we extend our hardness result to include the relaxation allowing redundant cuts [Subsection 8.2](#).

## 8.1 Making Almost Every Agent Envy-Free

In this section, we assume that there are  $p = k + 1$  agents and  $k$  cuts. Additionally, we assume that each agent is allocated exactly one piece of the cake.

**Definition 68** ((3-out-of- $k$ ,  $\varepsilon$ )-Mostly Approximately Envy-Free Cake Cut). *Let  $\mathbf{x}$  be a  $k$ -cut. We say  $\mathbf{x}$  is a (3-out-of- $k$ ,  $\varepsilon$ )-mostly approximately envy-free cake cut if there exists a permutation  $\pi : [p] \rightarrow [p]$  and a subset of agents  $S \subseteq [p]$  consisting of at least three agents such that for each agent  $d \in S$ , it holds  $u_d(\mathbf{x}, X_{\pi(d)}) + \varepsilon \geq u_d(\mathbf{x}, X_i)$  for any  $i$ .*

Our approach involves constructing the utility function and preference list for the (3-out-of- $k$ ,  $\varepsilon$ )-mostly approximately envy-free cake cut problem from the hard instance of 3-out-of- $k+1$  Approximate Symmetric SPERNER (Theorem 15). Then, we show that if we can find a solution for the (3-out-of- $k$ ,  $\varepsilon$ )-mostly approximately envy-free cake cut problem, the corresponding point representing the  $k$ -cut lies within a base simplex with at least three colors in the hard instance of Theorem 15, indicating the problem's PPAD-hardness. It is important to note that we use the same preference for each agent, denoted as  $P$ , and their utility as  $u$ .

### Obtaining preference and utility from the simplex colors:

- **Preference and Utility on discrete points:** Let  $N = 2^n - 1$ . For  $\mathbf{x} \in \Delta_n^k$ , we let  $P(\mathbf{x}) = C(\mathbf{x})$ , i.e. the preference of agents for  $k$ -cut  $\mathbf{x}$  is the color of point  $\mathbf{x}$  in the hard instance of Theorem 15. We first define a *pseudo-utility*  $u'$ , i.e. the value of a piece depends on the entire  $k$ -cut instead of the equivalent class of the  $k$ -cut. We use  $u'(\mathbf{x}, X_i)$  to denote the pseudo-utility of piece  $X_i$ . We let  $u'(\mathbf{x}, X_{P(\mathbf{x})}) = 1/(2N)$ . Also, if  $X_i$  is empty, we let  $u'(\mathbf{x}, X_i) = 0$ . For all other pieces  $X_j$ , we let  $u'(\mathbf{x}, X_j) = 1/(10k^2N)$ .
- **Interpolation:** Up to this point, we have only selected preferences and utilities for points whose coordinates are multiples of  $1/N$ , i.e. nodes in the discretized simplex. Let  $\mathbf{x}$  be a point within a base simplex with corners  $\mathbf{x}^{(0)}, \mathbf{x}^{(1)}, \dots, \mathbf{x}^{(k)}$ . Since  $\mathbf{x}$  is inside this base simplex, it can be written uniquely as a convex combination of its corners, i.e.  $\mathbf{x} = \sum_{i=0}^k \alpha_i \cdot \mathbf{x}^{(i)}$  such that  $\sum_{i=0}^k \alpha_i = 1$ . For piece  $X_j$ , we let  $u'(\mathbf{x}, X_j) = \sum_{i=0}^k \alpha_i \cdot u'(\mathbf{x}^{(i)}, X_j^{(i)})$ . Finally, for an equivalent class  $[\mathbf{x}]$  we let  $u([\mathbf{x}], X_i) = u'(\mathbf{x}, X_i)$  where  $\mathbf{x}$  is an arbitrary vector in this equivalent class. We will later (statement in Property 73 and proved in Lemma 75) show that for two  $k$ -cuts that are equivalent, the pseudo-utility of the similar pieces is going to be equal anyway. Now, we prove that the utility that we defined has both the non-negativity and Lipschitz conditions.

**Claim 69.** *The defined utility function  $u$  is Lipschitz.*

*Proof.* Consider the  $j$ -th piece for two distinct points  $\mathbf{x}, \mathbf{y} \in \Delta_n^k$ . First, we have  $\|\mathbf{x} - \mathbf{y}\|_1 \geq 1/N$ . Also,  $|u([\mathbf{x}], X_j) - u([\mathbf{y}], Y_j)| \leq 1/(2N)$  since the utility of a piece cannot be larger than  $1/(2N)$  by the definition of the utility function. Hence, we have  $|u([\mathbf{x}], X_j) - u([\mathbf{y}], Y_j)| \leq \|\mathbf{x} - \mathbf{y}\|_1$ . Now we prove the claim for two points within the same base simplex. Let  $\mathbf{x}$  and  $\mathbf{y}$  be two points within the base simplex with corners  $\mathbf{z}^{(0)}, \mathbf{z}^{(1)}, \dots, \mathbf{z}^{(k)}$ . Thus,  $\mathbf{x} = \sum_{i=0}^k \alpha_i \mathbf{z}^{(i)}$  and  $\mathbf{y} = \sum_{i=0}^k \beta_i \mathbf{z}^{(i)}$  s.t.  $\sum_{i=0}^k \alpha_i = \sum_{i=0}^k \beta_i = 1$ . For the  $j$ -th piece,

$$\begin{aligned} \left| u([\mathbf{x}], X_j) - u([\mathbf{y}], Y_j) \right| &= \left| \sum_{i=0}^k \alpha_i u([\mathbf{z}^{(i)}], X_j^{(i)}) - \sum_{i=0}^k \beta_i u([\mathbf{z}^{(i)}], X_j^{(i)}) \right| = \left| \sum_{i=0}^k u([\mathbf{z}^{(i)}], X_j^{(i)}) \cdot (\alpha_i - \beta_i) \right| \\ &\leq \sum_{i=0}^k u([\mathbf{z}^{(i)}], X_j^{(i)}) \cdot |\alpha_i - \beta_i| \\ &\leq \frac{1}{2N} \sum_{i=0}^k |\alpha_i - \beta_i|. \end{aligned}$$

Since points  $\mathbf{z}^{(0)}, \mathbf{z}^{(1)}, \dots, \mathbf{z}^{(k)}$  are corners of a base simplex  $\Delta_n^k$ , without loss of generality we can assume that there exists a permutation  $\pi \in \mathcal{S}_k$  such that  $\mathbf{z}^{(i)} = \mathbf{z}^{(0)} + \sum_{\ell=1}^i 1/N \cdot \mathbf{e}_{\pi(\ell)}$ . Then, we have

$$\begin{aligned}
\|\mathbf{x} - \mathbf{y}\|_1 &= \sum_{\ell=1}^k \left| \sum_{i=0}^k \alpha_i z_{\ell}^{(i)} - \sum_{i=0}^k \beta_i z_{\ell}^{(i)} \right| = \sum_{\ell=1}^k \left| \sum_{i=0}^k z_{\ell}^{(i)} (\alpha_i - \beta_i) \right| \\
&= \sum_{\ell=1}^k \left| \sum_{i=0}^k z_{\pi(\ell)}^{(i)} (\alpha_i - \beta_i) \right| \\
&= \sum_{\ell=1}^k \left| \sum_{i=0}^{\ell-1} z_{\pi(\ell)}^{(0)} (\alpha_i - \beta_i) + \sum_{i=\ell}^k (z_{\pi(\ell)}^{(0)} + \frac{1}{N}) \cdot (\alpha_i - \beta_i) \right| \\
&= \sum_{\ell=1}^k \left| z_{\pi(\ell)}^{(0)} \sum_{i=0}^k (\alpha_i - \beta_i) + \frac{1}{N} \sum_{i=\ell}^k (\alpha_i - \beta_i) \right| \\
&= \frac{1}{N} \sum_{\ell=1}^k \left| \sum_{i=\ell}^k (\alpha_i - \beta_i) \right|,
\end{aligned}$$

where the last equality is followed by  $\sum_{i=0}^k (\alpha_i - \beta_i) = 0$ . Further,

$$\begin{aligned}
\sum_{\ell=0}^k |\alpha_{\ell} - \beta_{\ell}| &= \sum_{\ell=0}^k \left| \sum_{i=\ell}^k (\alpha_i - \beta_i) - \sum_{i=\ell+1}^k (\alpha_i - \beta_i) \right| \leq \sum_{\ell=0}^k \left| \sum_{i=\ell}^k (\alpha_i - \beta_i) \right| + \left| \sum_{i=\ell+1}^k (\alpha_i - \beta_i) \right| \\
&\leq \sum_{\ell=0}^k \left| 2 \sum_{i=\ell}^k (\alpha_i - \beta_i) \right| \\
&= 2 \sum_{\ell=1}^k \left| \sum_{i=\ell}^k (\alpha_i - \beta_i) \right|
\end{aligned}$$

where the last equality is followed by the fact that for  $\ell = 0$ , we have  $\sum_{i=\ell}^k (\alpha_i - \beta_i) = 0$ . Combining the above bounds, we obtain

$$\left| u([\mathbf{x}], X_j) - u([\mathbf{y}], Y_j) \right| \leq \frac{1}{2N} \sum_{i=0}^k |\alpha_i - \beta_i| \leq \frac{1}{N} \sum_{\ell=1}^k \left| \sum_{i=\ell}^k \alpha_i - \beta_i \right| = \|\mathbf{x} - \mathbf{y}\|_1,$$

which completes the proof for two points  $\mathbf{x}$  and  $\mathbf{y}$  within the same base simplex. For two points  $\mathbf{x}$  and  $\mathbf{y}$  that are not in the same base simplex, suppose that there are  $r > 1$  base simplices between them (including simplices that  $\mathbf{x}$  and  $\mathbf{y}$  belongs to) to reach  $\mathbf{y}$  from  $\mathbf{x}$  in the closest path. Hence, there exist points  $\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \dots, \mathbf{z}^{(r-1)}$  such that point  $\mathbf{z}^{(i)}$  is on both the  $i$ -th and  $(i+1)$ -th simplices and

$$\|\mathbf{x} - \mathbf{y}\|_1 = \sum_{i=1}^r \left\| \mathbf{z}^{(i-1)} - \mathbf{z}^{(i)} \right\|_1,$$

if we define  $\mathbf{z}^{(0)} = \mathbf{x}$  and  $\mathbf{z}^{(r)} = \mathbf{y}$ . Moreover, since  $\mathbf{z}^{(i)}$  and  $\mathbf{z}^{(i+1)}$  are within the same base simplex, we have

$|u([\mathbf{z}^{(i)}], Z_j^i) - u([\mathbf{z}^{(i+1)}], Z_j^i)| \leq \|\mathbf{z}^{(i)} - \mathbf{z}^{(i+1)}\|_1$ . Therefore,

$$\begin{aligned} |u([\mathbf{x}], X_j) - u([\mathbf{y}], Y_j)| &= \left| \sum_{i=1}^r u([\mathbf{z}^{(i)}], Z_j^i) - u([\mathbf{z}^{(i-1)}], Z_j^{(i-1)}) \right| \\ &\leq \sum_{i=1}^r |u([\mathbf{z}^{(i)}], Z_j^i) - u([\mathbf{z}^{(i-1)}], Z_j^{(i-1)})| \\ &\leq \sum_{i=1}^r \|\mathbf{z}^{(i)} - \mathbf{z}^{(i-1)}\|_1 \\ &= \|\mathbf{x} - \mathbf{y}\|_1, \end{aligned}$$

which concludes the proof.  $\square$

**Claim 70.** *The defined utility function  $u$  for the (3-out-of- $k$ ,  $\varepsilon$ )-mostly approximately envy-free cake cut problem satisfies the nonnegativity condition.*

*Proof.* For points  $\mathbf{x} \in \Delta_n^k$ , the utility of a piece is equal to zero if and only if it is an empty piece according to the way that we define the utility function. Now let  $\mathbf{x} \in \Delta^k$  and  $\mathbf{x}^{(0)}, \mathbf{x}^{(1)}, \dots, \mathbf{x}^{(k)}$  be the corner of base simplex that it belongs to. So we have  $\mathbf{x} = \sum_{i=0}^k \alpha_i \mathbf{x}^{(i)}$  s.t.  $\sum_{i=0}^k \alpha_i = 1$ .

Let  $i_1, \dots, i_r$  be the indices such that the  $j$ -th piece of  $\mathbf{x}^{(i_1)}, \dots, \mathbf{x}^{(i_r)}$  is empty, i.e. for each  $i \in \{i_1, \dots, i_r\}$ , we have  $x_j^{(i)} = x_{j-1}^{(i)}$  since the  $j$ -th piece is empty. Moreover,

$$\begin{aligned} x_j - x_{j-1} &= \sum_{i=0}^k \alpha_i x_j^{(i)} - \sum_{i=0}^k \alpha_i x_{j-1}^{(i)} = \sum_{i \in \{i_1, \dots, i_r\}} \alpha_i x_j^{(i)} + \sum_{i \notin \{i_1, \dots, i_r\}} \alpha_i x_j^{(i)} - \sum_{i \in \{i_1, \dots, i_r\}} \alpha_i x_{j-1}^{(i)} - \sum_{i \notin \{i_1, \dots, i_r\}} \alpha_i x_{j-1}^{(i)} \\ &= \left( \sum_{i \in \{i_1, \dots, i_r\}} \alpha_i (x_j^{(i)} - x_{j-1}^{(i)}) \right) + \left( \sum_{i \notin \{i_1, \dots, i_r\}} \alpha_i (x_j^{(i)} - x_{j-1}^{(i)}) \right). \end{aligned}$$

Note that in second term, we have  $x_j^{(i)} \neq x_{j-1}^{(i)}$  which implies that if there exists a non-zero  $\alpha_i$  in the second term, then  $x_j - x_{j-1} > 0$ . Thus,  $j$ -th piece is empty if and only if there exists no  $i \in \{0, \dots, k\} \setminus \{i_1, \dots, i_r\}$  where  $\alpha_i > 0$ . Further,

$$\begin{aligned} u([\mathbf{x}], X_j) &= \sum_{i=0}^k \alpha_i \cdot u([\mathbf{x}^{(i)}], X_j^{(i)}) = \left( \sum_{i \in \{i_1, \dots, i_r\}} \alpha_i \cdot u([\mathbf{x}^{(i)}], X_j^{(i)}) \right) + \left( \sum_{i \notin \{i_1, \dots, i_r\}} \alpha_i \cdot u([\mathbf{x}^{(i)}], X_j^{(i)}) \right) \\ &= \left( \sum_{i \notin \{i_1, \dots, i_r\}} \alpha_i \cdot u([\mathbf{x}^{(i)}], X_j^{(i)}) \right), \end{aligned}$$

where we have the last equality because  $u([\mathbf{x}^{(i)}], X_j^{(i)}) = 0$  for  $i \in \{i_1, \dots, i_r\}$ . On the other hand,  $u([\mathbf{x}^{(i)}], X_j^{(i)}) > 0$  for  $i \in \{i_1, \dots, i_r\}$ . Finally, since  $j$ -th piece is empty if and only if there exists no  $i \in \{0, \dots, k\} \setminus \{i_1, \dots, i_r\}$  where  $\alpha_i > 0$ , we have the utility of piece  $X_j$  is equal to zero if and only if it is empty.  $\square$

The way that we defined our utility functions enables us to show that if corners of a base simplex are colored with at most two colors, then none of the points inside this base simplex is a solution for the (3-out-of- $k$ ,  $\varepsilon$ )-mostly approximately envy-free cake cut problem. As a corollary, if we can find a (3-out-of- $k$ ,  $\varepsilon$ )-mostly approximately envy-free cake cut, the corresponding point in the simplex must be within a base simplex that has at least three colors.

**Lemma 71.** Let  $\mathbf{x} \in \Delta^k$  be a  $k$ -cut that is within a base simplex with corners  $\mathbf{x}^{(0)}, \mathbf{x}^{(1)}, \dots, \mathbf{x}^{(k)} \in \Delta_n^k$ . If

$$\left| \{P(\mathbf{x}^{(0)}), P(\mathbf{x}^{(1)}), \dots, P(\mathbf{x}^{(k)})\} \right| \leq 2,$$

then  $\mathbf{x}$  is not a solution for (3-out-of- $k$ ,  $\varepsilon$ )-mostly approximately envy-free cake cut problem when  $\varepsilon \leq 1/(10N)$ .

*Proof.* We can write  $\mathbf{x} = \sum_{i=0}^k \alpha_i \mathbf{x}^{(i)}$  s.t.  $\sum_{i=0}^k \alpha_i = 1$ . Let  $\mathcal{P} = \{P(\mathbf{x}^{(0)}), P(\mathbf{x}^{(1)}), \dots, P(\mathbf{x}^{(k)})\}$  and  $\bar{\mathcal{P}} = \{0, 1, \dots, k\} \setminus \mathcal{P}$ . For  $j' \in \bar{\mathcal{P}}$ , we have  $u([\mathbf{x}^{(i)}], X_{j'}^{(i)}) \leq 1/(10k^2N)$  by the definition of our utility function. Thus,

$$u([\mathbf{x}], X_{j'}) = \sum_{i=0}^k \alpha_i \cdot u([\mathbf{x}^{(i)}], X_{j'}^{(i)}) \leq \frac{1}{10k^2N} \sum_{i=0}^k \alpha_i = \frac{1}{10k^2N}.$$

On the other hand, since  $|\mathcal{P}| \leq 2$ , by pigeonhole principle there exists a  $j \in \mathcal{P}$  such that  $\sum_{i \text{ and } P(\mathbf{x}^{(i)})=j} \alpha_i \geq 1/2$  which implies that

$$\begin{aligned} u([\mathbf{x}], X_j) &= \sum_{i=0}^k \alpha_i \cdot u([\mathbf{x}^{(i)}], X_j^{(i)}) \geq \sum_{i \text{ and } P(\mathbf{x}^{(i)})=j} \alpha_i \cdot u([\mathbf{x}^{(i)}], X_j^{(i)}) = \sum_{i \text{ and } P(\mathbf{x}^{(i)})=j} \alpha_i \cdot u([\mathbf{x}^{(i)}], X_{P(\mathbf{x}^{(i)})}^{(i)}) \\ &= \frac{1}{2N} \cdot \sum_{i \text{ and } P(\mathbf{x}^{(i)})=j} \alpha_i \\ &\geq \frac{1}{4N}, \end{aligned}$$

where the last equality follows by the fact that  $u([\mathbf{x}^{(i)}], X_{P(\mathbf{x}^{(i)})}^{(i)}) = 1/(2N)$ . For any  $j' \in \bar{\mathcal{P}}$ , by combining the last two bounds for the specific  $j$  in the last bound, we obtain

$$u([\mathbf{x}], X_{j'}) + \frac{1}{10N} \leq \frac{1}{10k^2N} + \frac{1}{10N} < \frac{1}{4N} \leq u([\mathbf{x}], X_j).$$

Therefore, when  $\varepsilon \leq 1/(10N)$ , there exists a piece in  $\mathcal{P}$  whose utility is greater than that of pieces in  $\bar{\mathcal{P}}$  by a margin larger than  $\varepsilon$ . Consequently, as  $|\mathcal{P}| \leq 2$ , there are at most two pieces where agents do not envy each other if we ignore the  $\varepsilon$  gap in their utilities. This implies that  $\mathbf{x}$  cannot be a solution for the (3-out-of- $k$ ,  $\varepsilon$ )-mostly approximately envy-free cake cut problem when  $\varepsilon \leq 1/(10N)$ .  $\square$

## Properties of the Utility Function

Note that when constructing utilities and preferences, we must satisfy several properties to accurately model our cake-cutting problem. The first property is the *boundary property*, which means that for a point  $\mathbf{x}$ , if  $X_i$  represents an empty piece, then the utility of that piece should be equal to zero. It is easy to see that this property is satisfied due to [Claim 70](#).

**Property 72 (Boundary).** Let  $\mathbf{x} \in \Delta^k$  be a  $k$ -cut and  $X = (X_0, X_1, \dots, X_k)$  be its corresponding pieces of cake. If  $X_i$  is an empty piece of cake, then  $u([\mathbf{x}], X_i) = 0$  and  $P(\mathbf{x}) \neq i$ .

The second property that needs to be satisfied is that the facets of the simplex should resemble one another under permutations of colors. This property is referred to as the *symmetry property*. To see why such a property is necessary, let us consider the case with 3 agents and 2 cuts. Consider two distinct vectors,  $\mathbf{x} = (0, 0.5)$  and  $\mathbf{y} = (0.5, 1)$ , which correspond to two different sets of 2-cuts. It is important to note

that the pieces obtained from  $\mathbf{x}$  and  $\mathbf{y}$  are exactly alike, but their order differs, i.e.,  $X_1 = Y_0$  and  $X_2 = Y_1$ . Consequently, it must hold that  $u'(\mathbf{x}, X_1) = u'(\mathbf{y}, Y_0)$  and  $u'(\mathbf{x}, X_2) = u'(\mathbf{y}, Y_1)$ . From the preference viewpoint, if we have  $P(\mathbf{x}) = 1$ , then  $P(\mathbf{y}) = 0$  due to this similarity. As a result of this property, for two  $k$ -cuts that are equivalent, the utility of a similar piece is equal.

**Property 73** (Symmetry). *Let  $\mathbf{x}, \mathbf{y} \in \Delta^k$  be two  $k$ -cuts and  $X = (X_0, X_1, \dots, X_k)$  and  $Y = (Y_0, Y_1, \dots, Y_k)$  be their corresponding pieces of cake. Suppose that there exists a permutation  $\pi : \{0, \dots, k\} \rightarrow \{0, \dots, k\}$  such that  $X_i = Y_{\pi(i)}$  for all  $i \in \{0, \dots, k\}$ . Then,  $u'(\mathbf{x}, X_i) = u'(\mathbf{y}, Y_{\pi(i)})$  for each  $i$ .*

Any coloring of simplex that we use to construct our utility function and preference must satisfy the symmetry property (Property 73). It is worth noting that when aiming to achieve  $\varepsilon$ -approximate envy-free cake cutting for all agents, similar to [DQS12], this property loses its significance. This is because in the facets of the simplex, there exists no solution, and therefore satisfying this property cannot hurt the hard instance. On the other hand, in the (3-out-of- $k$ ,  $\varepsilon$ )-mostly approximately envy-free cake cut, there indeed exists a solution on the facets of the simplex. Therefore, we must be careful in how we color the simplex to ensure it satisfies the symmetry property. In Lemma 75, we show that our utility function that is obtained from the hard instance in Theorem 15 satisfies the symmetry property.

**Claim 74.** *Let  $\mathbf{x}, \mathbf{y} \in \Delta_n^k$  be two  $k$ -cuts in a  $k$ -dimensional simplex that satisfies the symmetric condition in Definition 14. Let  $i \in [k]$  and suppose that  $X_j = Y_j$  for  $j \in \{0, \dots, k\} \setminus \{i-1, i\}$ . Moreover, let  $X_{i-1} = Y_i$ ,  $X_i = Y_{i-1}$ , and  $X_{i-1}$  be an empty piece. Then,  $P(\mathbf{y}) = P(\mathbf{x})$  if  $P(\mathbf{x}) \neq i$ , and  $P(\mathbf{y}) = i-1$ , otherwise.*

*Proof.* Note that  $\mathbf{x}$  and  $\mathbf{y}$  satisfy the symmetric condition in Definition 14 by the definition. This is because all pieces except the  $i$ -th and  $(i-1)$ -th pieces are exactly similar, with one of them being empty. Consequently, the claim holds true.  $\square$

**Lemma 75.** *The proposed utility function  $u$  for the (3-out-of- $k$ ,  $\varepsilon$ )-mostly approximately envy-free cake cut problem satisfies the symmetric condition.*

*Proof.* Note that it suffices to show that the utility function satisfies the symmetry property solely for points in  $\Delta_n^k$ . Subsequently, as the pseudo-utilities are a convex combination of the corners of base simplices, it holds for all points in  $\Delta^k$ . Further, it is enough to show that if  $P(\mathbf{x}) = i$ , then  $P(\mathbf{y}) = \pi(i)$ . This is because the pseudo-utility of empty pieces is zero in both cuts, the pseudo-utility of the preferred piece is  $1/(2N)$ , and all other non-empty pieces have a pseudo-utility of  $1/(10k^2N)$  for points in  $\Delta_n^k$ .

Let  $\mathbf{x}, \mathbf{y} \in \Delta_n^k$  where there exists a permutation  $\pi : \{0, \dots, k\} \rightarrow \{0, \dots, k\}$  such that  $X_i = Y_{\pi(i)}$  for all  $i \in \{0, \dots, k\}$ . First, by the boundary property, the preference of agents cannot be an empty piece. Thus, we only need to show that for permutations that map the non-empty pieces from one cut to the other cut, the preference of agents does not change. Let  $X_{i_0}, X_{i_1}, \dots, X_{i_r}$  be the non-empty pieces of  $X$  such that  $i_0 < i_1 < \dots < i_r$ . Similarly, let  $Y_{j_0}, Y_{j_1}, \dots, Y_{j_r}$  be the non-empty pieces of  $Y$  such that  $j_0 < j_1 < \dots < j_r$ . Note that for permutation  $\pi$  such that  $\pi(i_d) = j_d$  and arbitrary maps empty pieces to each other, we have  $X_i = Y_{\pi(i)}$  for all  $i \in \{0, \dots, k\}$ . Now, we need to show that if  $P(\mathbf{x}) = i_d$ , then  $P(\mathbf{y}) = \pi(i_d)$  since empty pieces cannot be a preferred piece.

Let  $\mathbf{z} \in \Delta_n^k$  to be a point in the simplex which has this property that  $Z_d$  is empty if  $d > r$  and otherwise,  $Z_d = X_{i_d} = Y_{j_d}$ . Basically, empty pieces are the rightmost pieces in  $Z$ . We claim that if  $P(\mathbf{x}) = i_d$  for some  $d \leq r$ , then  $P(\mathbf{z}) = d$ . We prove this claim step by step. In each step, we swap two consecutive pieces where the left piece is empty and the right piece is not empty. Note that by doing this process a finite number of times, we will finally end up with  $Z$  where all empty pieces are in the rightmost positions. Let  $X^{(0)}, X^{(1)}, \dots, X^{(t)}$  be the different configurations of the pieces that we see in this process and  $\mathbf{x}^{(0)}, \mathbf{x}^{(1)}, \dots, \mathbf{x}^{(t)}$  be their corresponding  $k$ -cuts vectors. So we have  $X^{(0)} = X$  and  $X^{(t)} = Z$ . Note that  $P(\mathbf{x}^{(0)}) = i_d$  and every time that we convert  $X^{(i)}$  to  $X^{(i+1)}$ , if we swap  $P(\mathbf{x}^{(i)})$  and an empty piece, then we

have  $P(\mathbf{x}^{(i+1)}) = P(\mathbf{x}^{(i)}) - 1$  by [Claim 74](#). We call this type of swap as *critical swap*. By the definition of  $Z$ , during this process, we have exactly  $i_d - d$  critical swaps. Therefore, we have

$$P(\mathbf{z}) = P(\mathbf{x}^{(i)}) = P(\mathbf{x}^{(0)}) - (i_d - d) = P(\mathbf{x}) - (i_d - d) = d,$$

which completes the proof of the claim. With a similar argument, we can show that if  $P(\mathbf{z}) = d$  for some  $d \leq r$ , then  $P(\mathbf{y}) = j_d$ . Now assume that  $P(\mathbf{x}) = i_d$ . By the above claim, we have  $P(\mathbf{z}) = d$  and consequently,  $P(\mathbf{y}) = j_d$ . Since  $\pi(i_d) = j_d$ , we have  $P(\mathbf{y}) = \pi(i_d)$  which conclude the proof.  $\square$

Now we are ready to prove our main theorem for the (3-out-of- $k$ ,  $\epsilon$ )-mostly approximately envy-free cake cut problem.

**Theorem 76.** *For any constant  $\epsilon$  such that  $k < \log^{1-\delta}(1/\epsilon)$  for some constant  $\delta > 0$ , the problem of finding the (3-out-of- $k$ ,  $\epsilon$ )-mostly approximately envy-free cake cut for  $k$  cuts and  $k + 1$  agents is PPAD-complete. Further, it requires a query complexity of  $(1/\epsilon)^{\Omega(1/k)}/\text{polylog}(1/\epsilon)$ .*

*Proof.* We assume that the utilities of all agents are as defined at the beginning of this subsection. By [Claim 69](#) and [Claim 70](#), the Lipschitz and nonnegativity conditions hold for utility functions. Further, by [Claim 70](#) and [Lemma 75](#), we have both the boundary and symmetry properties ([Property 72](#) and [Property 73](#)). Hence, the utility function is a valid utility function for the cake-cutting problem.

We defined our utilities and preferences based on the circuit  $C : \Delta_n^k \rightarrow \{0, \dots, k\}$  that is constructed in [Theorem 15](#). We know that  $\epsilon$ -approximate 3-out-of- $p$ -envy-free cake cut problem is in PPAD by [\[DQS12\]](#). If we can find an allocation that is a solution for the (3-out-of- $k$ ,  $\epsilon$ )-mostly approximately envy-free cake cut, we can convert the allocation to its corresponding point in the simplex. Further, by [Lemma 71](#), if we can find a solution for (3-out-of- $k$ ,  $\epsilon$ )-mostly approximately envy-free cake cut when  $\epsilon \leq 1/(10N)$ , then the corresponding point in the simplex is in a trichromatic base simplex. Thus, we can find a solution for the 3-out-of- $k+1$  Approximate Symmetric SPERNER problem. By [Theorem 15](#), 3-out-of- $k+1$  Approximate Symmetric SPERNER is PPAD-hard, and therefore, the (3-out-of- $k$ ,  $\epsilon$ )-mostly approximately envy-free cake cut problem is PPAD-hard.

To get the query complexity, consider  $N = 2^{4kn}$ . Thus, we have  $nk = \Theta(\log N)$  and  $n = \Theta(\log N/k)$ . Plugging  $k < \log^{1-\delta}(N)$ , we have  $n = \text{polylog}(N)$  and  $k = \text{poly}(n)$ . Also, we have  $N = \Theta(1/\epsilon)$ . Therefore, by [Theorem 15](#), the query complexity is at least  $(1/\epsilon)^{\Omega(1/k)}/\text{polylog}(1/\epsilon)$ .  $\square$

## 8.2 Beyond Connected Pieces

In this section, we consider both relaxations: (1) finding three agents who do not envy others, and (2) allowing agents to have more than one piece. More formally, we have  $p$  agents where  $p \leq k + 1$ , and our objective is to find a bundling of  $k + 1$  pieces into  $p$  bundles. The goal is to allocate each agent one of these bundles in such a way that the allocation is  $\epsilon$ -approximate envy-free for at least three agents.

**Definition 19** ( $\epsilon$ -Approximate  $p'$ -out-of- $p$ -Envy-Free Cake Cut). *Given a  $k$ -cut  $\mathbf{x}$ , a bundling  $\mathcal{B}$  of the induced cake pieces, an assignment  $\pi : [p] \rightarrow [p]$  of the bundles to agents,  $\epsilon > 0$ , and  $p' < p$  we say that  $(\mathbf{x}, \mathcal{B}, \pi)$  is  $\epsilon$ -approximate  $p'$ -out-of- $p$ -envy-free if there exists a subset  $S \subseteq [p]$  of  $p'$  agents, such that for every  $d \in S$ ,  $u_d([\mathbf{x}], \mathcal{B}_{\pi(d)}) + \epsilon \geq u_d([\mathbf{x}], \mathcal{B}_i)$  for every bundle  $\mathcal{B}_i$ .*

Our approach for this subsection is very similar to the previous one. We first extend the utility function for multiple pieces. Then, we demonstrate that if we can find a solution for the  $\epsilon$ -approximate 3-out-of- $p$ -envy-free cake cut, the corresponding point representing the  $k$ -cut lies within a base simplex with at least three colors in the hard instance of [Theorem 15](#), indicating the problem's PPAD-hardness.



**Extending the utility function to multiple pieces:** We let  $u'(\mathbf{x}, B)$  be equal to the sum of the utilities of the pieces that belong to bundle  $B$ . Since we defined the pseudo-utility function the same as the previous section, for two  $k$ -cuts that are equivalent, the pseudo-utility of the similar pieces is going to be equal with a similar argument (formalized in [Property 82](#) and [Claim 83](#)). Finally, we let  $u([\mathbf{x}], B)$  be equal to the sum of the utilities of the pieces that belong to bundle  $B$ . Since the utility function is defined the same way as the previous subsection, we have both the Lipschitz and nonnegativity conditions. We repeat the statements for the usage of this subsection.

**Claim 77.** *The defined utility function  $u$  for the  $\varepsilon$ -approximate 3-out-of- $p$ -envy-free cake cut problem is Lipschitz.*

**Claim 78.** *The defined utility function  $u$  for the  $\varepsilon$ -approximate 3-out-of- $p$ -envy-free cake cut problem satisfies the nonnegativity condition.*

We now show that for the extended utility function, any cut that is an  $\varepsilon$ -approximate 3-out-of- $p$ -envy-free cake cut implies that the corresponding point in the simplex must lie within a trichromatic base simplex.

**Lemma 79.** *Let  $\mathbf{x} \in \Delta^k$  be a  $k$ -cut that is within a base simplex with corners  $\mathbf{x}^{(0)}, \mathbf{x}^{(1)}, \dots, \mathbf{x}^{(k)} \in \Delta_n^k$ . If*

$$\left| \{P(\mathbf{x}^{(0)}), P(\mathbf{x}^{(1)}), \dots, P(\mathbf{x}^{(k)})\} \right| \leq 2,$$

*then, there exists no bundling for the point  $\mathbf{x}$  that serves as a solution for the  $\varepsilon$ -approximate 3-out-of- $p$ -envy-free cake cut problem when  $\varepsilon \leq 1/(10N)$ .*

*Proof.* We use almost the same approach as proof of [Lemma 71](#). We can write  $\mathbf{x} = \sum_{i=0}^k \alpha_i \mathbf{x}^{(i)}$  s.t.  $\sum_{i=0}^k \alpha_i = 1$ . Let  $\mathcal{P} = \{P(\mathbf{x}^{(0)}), P(\mathbf{x}^{(1)}), \dots, P(\mathbf{x}^{(k)})\}$  and  $\overline{\mathcal{P}} = \{0, 1, \dots, k\} \setminus \mathcal{P}$ . For  $j' \in \overline{\mathcal{P}}$ , with the same argument as the proof of [Lemma 71](#), we have  $u([\mathbf{x}], X_{j'}) \leq 1/(10k^2N)$ . Therefore, if a bundle  $B_{d'}$  only contains pieces that are in  $\overline{\mathcal{P}}$ , we have that  $u([\mathbf{x}], B_{d'}) \leq (k+1)/(10k^2N)$  since the bundle has at most  $k+1$  of these pieces and each of them has a utility of at most  $1/(10k^2N)$ .

On the other hand, since  $|\mathcal{P}| \leq 2$ , by pigeonhole principle there exists a  $j \in \mathcal{P}$  such that  $\sum_{i \text{ and } P(\mathbf{x}^{(i)})=j} \alpha_i \geq 1/2$  which implies that  $u(\mathbf{x}, X_j) \geq 1/(4N)$  for the same  $j$ , again using the same argument as [Lemma 71](#). Now consider bundle  $B_d$  that contains this piece. We have that  $u([\mathbf{x}], B_d) \geq 1/(4N)$ .

As  $|\mathcal{P}| \leq 2$ , at most two bundles can have a utility larger than or equal to  $1/(4N)$ . By the above argument, there exists a bundle with a utility of at least  $1/(4N)$ . Additionally, there exist  $p-2$  bundles, each with a utility of at most  $(k+1)/(10k^2N)$ . Hence, when  $\varepsilon \leq 1/(10N)$ , this implies that  $\mathbf{x}$  cannot be a solution for the  $\varepsilon$ -approximate 3-out-of- $p$ -envy-free cake cut problem, as at least  $p-2$  agents would envy others.  $\square$

We need to satisfy the boundary and symmetry properties when we have bundling similar to that in [Subsection 8.1](#). We redefine these two properties for the  $\varepsilon$ -approximate 3-out-of- $p$ -envy-free cake cut problem. Regarding the boundary property, any bundle of a  $k$ -cut vector that is empty must have a utility equal to zero.

**Property 80 (Boundary).** *Let  $\mathbf{x}$  be a  $k$ -cut and  $\mathcal{B} = \{B_0, B_1, \dots, B_{p-1}\}$  be a bundling of its pieces into  $p$  bundles. If  $B_i$  is empty, then  $u(\mathbf{x}, B_i) = 0$ .*

**Claim 81.** *The proposed utility function  $u$  satisfies the boundary property.*

*Proof.* Let  $B_i$  be an empty bundle consisting of pieces  $X_{i_1}, \dots, X_{i_r}$ . Since  $B_i$  is empty, all pieces  $X_{i_1}, \dots, X_{i_r}$  are empty pieces. Thus,

$$u([\mathbf{x}], B_i) = \sum_{j=1}^r u([\mathbf{x}], X_{i_j}) = 0,$$

where the last equality is followed by [Claim 78](#).  $\square$

The symmetry property is also very similar to the symmetry property (Property 73) in Subsection 8.1, with slight modifications to suit our application in this section.

**Property 82** (symmetry). *Let  $\mathbf{x}$  and  $\mathbf{y}$  be two  $k$ -cuts. Suppose that there exists a permutation  $\pi : \{0, \dots, k\} \rightarrow \{0, \dots, k\}$  such that  $X_i = Y_{\pi(i)}$  for all  $i$ . Let  $\mathcal{B}^{(x)} = \{B_0^{(x)}, \dots, B_{p-1}^{(x)}\}$  and  $\mathcal{B}^{(y)} = \{B_0^{(y)}, \dots, B_{p-1}^{(y)}\}$  be bundling of the pieces  $X$  and  $Y$ , respectively. Suppose that  $B_i^{(x)} = \{X_{i_1}, \dots, X_{i_r}\}$  and  $B_j^{(y)} = \{Y_{\pi(i_1)}, \dots, Y_{\pi(i_r)}\}$ . Then,  $u'(\mathbf{x}, B_i^{(x)}) = u'(\mathbf{y}, B_j^{(y)})$ .*

**Claim 83.** *The proposed utility function  $u$  for the  $\varepsilon$ -approximate 3-out-of- $p$ -envy-free cake cut problem satisfies the symmetry property.*

*Proof.* Let  $\mathbf{x}$  and  $\mathbf{y}$  be two  $k$ -cuts with the condition in the premise of Property 82. Since we use the exact the same utility function as the (3-out-of- $k$ ,  $\varepsilon$ )-mostly approximately envy-free cake cut problem, by Lemma 75, we have  $u'(\mathbf{x}, X_i) = u'(\mathbf{y}, Y_{\pi(i)})$ . Therefore, we can conclude the proof since

$$u'(\mathbf{x}, B_i^{(x)}) = \sum_{d=1}^r u'(\mathbf{x}, X_{i_d}) = \sum_{d=1}^r u'(\mathbf{y}, Y_{\pi(i_d)}) = u'(\mathbf{y}, B_j^{(y)}). \quad \square$$

Now we are ready to prove our main theorem for the  $\varepsilon$ -Approximate 3-out-of- $p$ -envy-free cake cut problem.

**Theorem 20.** *For any constants  $3 \leq p \leq k+1$  and  $\varepsilon$  such that  $k < \log^{1-\delta}(1/\varepsilon)$  for some constant  $\delta > 0$ , the problem of finding an  $\varepsilon$ -approximate 3-out-of- $p$ -envy-free cake cut with  $k$  cuts is PPAD-complete. If, instead, the algorithm has black-box access to a value oracle, it requires a query complexity of  $(1/\varepsilon)^{\Omega(1/k)}/\text{polylog}(1/\varepsilon)$ .*

*Proof.* We assume that the utilities of all agents are as defined at the beginning of this subsection. By Claim 77 and Claim 78, the Lipschitz and non-negativity conditions for utility functions are satisfied. Moreover, according to Claim 81 and Claim 83, we observe that both the boundary and symmetry properties (Property 80 and Property 82) hold. Hence, the utility function is a valid utility function for the cake-cutting problem.

We have defined our utilities and preferences based on the circuit  $C : \Delta_n^k \rightarrow \{0, \dots, k\}$  constructed in Theorem 15. We know that  $\varepsilon$ -approximate 3-out-of- $p$ -envy-free cake cut problem is in PPAD by [DQS12]. Moreover, as stated in Lemma 79, if a solution for the  $\varepsilon$ -approximate 3-out-of- $p$ -envy-free cake cut problem is found when  $\varepsilon \leq 1/(10N)$ , then the corresponding point in the simplex lies within a trichromatic base simplex. Thus, we can solve the 3-out-of- $k+1$  Approximate Symmetric SPERNER problem. Since 3-out-of- $k+1$  Approximate Symmetric SPERNER is proven to be PPAD-hard by Theorem 15, it follows that the  $\varepsilon$ -approximate 3-out-of- $p$ -envy-free cake cut problem is also PPAD-hard.

To get the query complexity, consider  $N = 2^{4kn}$ . Thus, we have  $nk = \Theta(\log N)$  and  $n = \Theta(\log N/k)$ . Plugging  $k < \log^{1-\delta}(N)$ , we have  $n = \text{polylog}(N)$  and  $k = \text{poly}(n)$ . Also, we have  $N = \Theta(1/\varepsilon)$ . Therefore, by Theorem 15, the query complexity is at least  $(1/\varepsilon)^{\Omega(1/k)}/\text{polylog}(1/\varepsilon)$ .  $\square$

## Acknowledgement

This research was partially supported by NSF grants CCF-2112824, CCF-2209520, and CCF-2312156, ONR under award N000142212771, a Stanford Graduate Fellowship, Dantzig-Lieberman Operations Research Graduate Fellowship, and the David and Lucile Packard Foundation Fellowship.

## References

- [AAB+23] Georgios Amanatidis, Haris Aziz, Georgios Birmpas, Aris Filos-Ratsikas, Bo Li, Hervé Moulin, Alexandros A. Voudouris, and Xiaowei Wu. “Fair division of indivisible goods: Recent progress and open questions”. In: *Artif. Intell.* 322 (2023), p. 103965. DOI: [10.1016/J.ARTINT.2023.103965](https://doi.org/10.1016/J.ARTINT.2023.103965). URL: <https://doi.org/10.1016/j.artint.2023.103965>.
- [ABR19] Eshwar Ram Arunachaleswaran, Siddharth Barman, and Nidhi Rathi. “Fully Polynomial-Time Approximation Schemes for Fair Rent Division”. In: *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*. Ed. by Timothy M. Chan. SIAM, 2019, pp. 1994–2013. DOI: [10.1137/1.9781611975482.121](https://doi.org/10.1137/1.9781611975482.121). URL: <https://doi.org/10.1137/1.9781611975482.121>.
- [AM20] Haris Aziz and Simon Mackenzie. “A bounded and envy-free cake cutting algorithm”. In: *Commun. ACM* 63.4 (2020), pp. 119–126. DOI: [10.1145/3382129](https://doi.org/10.1145/3382129). URL: <https://doi.org/10.1145/3382129>.
- [BN19] Simina Brânzei and Noam Nisan. “Communication Complexity of Cake Cutting”. In: *Proceedings of the 2019 ACM Conference on Economics and Computation, EC 2019, Phoenix, AZ, USA, June 24-28, 2019*. Ed. by Anna Karlin, Nicole Immorlica, and Ramesh Johari. ACM, 2019, p. 525. DOI: [10.1145/3328526.3329644](https://doi.org/10.1145/3328526.3329644). URL: <https://doi.org/10.1145/3328526.3329644>.
- [BN22] Simina Brânzei and Noam Nisan. “The Query Complexity of Cake Cutting”. In: *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*. Ed. by Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh. 2022. URL: [http://papers.nips.cc/paper%5C\\_files/paper/2022/hash/f7a7bb369e48f10e85fce85b67d8c516-Abstract-Conference.html](http://papers.nips.cc/paper%5C_files/paper/2022/hash/f7a7bb369e48f10e85fce85b67d8c516-Abstract-Conference.html).
- [BPR16] Yakov Babichenko, Christos H. Papadimitriou, and Aviad Rubinfeld. “Can Almost Everybody be Almost Happy?” In: *Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science, Cambridge, MA, USA, January 14-16, 2016*. Ed. by Madhu Sudan. ACM, 2016, pp. 1–9. DOI: [10.1145/2840728.2840731](https://doi.org/10.1145/2840728.2840731). URL: <https://doi.org/10.1145/2840728.2840731>.
- [CD09] Xi Chen and Xiaotie Deng. “On the complexity of 2D discrete fixed point problem”. In: *Theor. Comput. Sci.* 410.44 (2009), pp. 4448–4456.
- [CGMM20] Bhaskar Ray Chaudhury, Jugal Garg, Peter McGlaughlin, and Ruta Mehta. “Dividing Bads is Harder than Dividing Goods: On the Complexity of Fair and Efficient Division of Chores”. In: *CoRR* abs/2008.00285 (2020). arXiv: [2008.00285](https://arxiv.org/abs/2008.00285). URL: <https://arxiv.org/abs/2008.00285>.
- [CPY17] Xi Chen, Dimitris Paparas, and Mihalis Yannakakis. “The Complexity of Non-Monotone Markets”. In: *J. ACM* 64.3 (2017), 20:1–20:56. DOI: [10.1145/3064810](https://doi.org/10.1145/3064810). URL: <https://doi.org/10.1145/3064810>.
- [DFH22] Argyrios Deligkas, Aris Filos-Ratsikas, and Alexandros Hollender. “Two’s company, three’s a crowd: Consensus-halving for a constant number of agents”. In: *Artif. Intell.* 313 (2022), p. 103784. DOI: [10.1016/J.ARTINT.2022.103784](https://doi.org/10.1016/J.ARTINT.2022.103784). URL: <https://doi.org/10.1016/j.artint.2022.103784>.

- [DFHM22] Argyrios Deligkas, John Fearnley, Alexandros Hollender, and Themistoklis Melissourgos. “Constant inapproximability for PPA”. In: *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing, Rome, Italy, June 20 - 24, 2022*. Ed. by Stefano Leonardi and Anupam Gupta. ACM, 2022, pp. 1010–1023. DOI: [10.1145/3519935.3520079](https://doi.org/10.1145/3519935.3520079). URL: <https://doi.org/10.1145/3519935.3520079>.
- [DGP09] Constantinos Daskalakis, Paul W. Goldberg, and Christos H. Papadimitriou. “The Complexity of Computing a Nash Equilibrium”. In: *SIAM J. Comput.* 39.1 (2009), pp. 195–259. DOI: [10.1137/070699652](https://doi.org/10.1137/070699652). URL: <https://doi.org/10.1137/070699652>.
- [DQS12] Xiaotie Deng, Qi Qi, and Amin Saberi. “Algorithmic Solutions for Envy-Free Cake Cutting”. In: *Oper. Res.* 60.6 (2012), pp. 1461–1476.
- [FG18] Aris Filos-Ratsikas and Paul W. Goldberg. “Consensus halving is PPA-complete”. In: *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*. Ed. by Ilias Diakonikolas, David Kempe, and Monika Henzinger. ACM, 2018, pp. 51–64. DOI: [10.1145/3188745.3188880](https://doi.org/10.1145/3188745.3188880). URL: <https://doi.org/10.1145/3188745.3188880>.
- [FHSZ21] Aris Filos-Ratsikas, Alexandros Hollender, Katerina Sotiraki, and Manolis Zampetakis. “A Topological Characterization of Modulo- $p$  Arguments and Implications for Necklace Splitting”. In: *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms, SODA 2021, Virtual Conference, January 10 - 13, 2021*. Ed. by Dániel Marx. SIAM, 2021, pp. 2615–2634. DOI: [10.1137/1.9781611976465.155](https://doi.org/10.1137/1.9781611976465.155). URL: <https://doi.org/10.1137/1.9781611976465.155>.
- [FHSZ23] Aris Filos-Ratsikas, Alexandros Hollender, Katerina Sotiraki, and Manolis Zampetakis. “Consensus-Halving: Does It Ever Get Easier?” In: *SIAM J. Comput.* 52.2 (2023), pp. 412–451. DOI: [10.1137/20M1387493](https://doi.org/10.1137/20M1387493). URL: <https://doi.org/10.1137/20M1387493>.
- [GH21] Paul W. Goldberg and Alexandros Hollender. “The Hairy Ball problem is PPAD-complete”. In: *J. Comput. Syst. Sci.* 122 (2021), pp. 34–62. DOI: [10.1016/J.JCSS.2021.05.004](https://doi.org/10.1016/j.jcss.2021.05.004). URL: <https://doi.org/10.1016/j.jcss.2021.05.004>.
- [GHH23] Paul W. Goldberg, Kasper Høgh, and Alexandros Hollender. “The Frontier of Intractability for EFX with Two Agents”. In: *Algorithmic Game Theory - 16th International Symposium, SAGT 2023, Egham, UK, September 4-7, 2023, Proceedings*. Ed. by Argyrios Deligkas and Aris Filos-Ratsikas. Vol. 14238. Lecture Notes in Computer Science. Springer, 2023, pp. 290–307. DOI: [10.1007/978-3-031-43254-5\\_17](https://doi.org/10.1007/978-3-031-43254-5_17). URL: [https://doi.org/10.1007/978-3-031-43254-5\\_17](https://doi.org/10.1007/978-3-031-43254-5_17).
- [HPV89] Michael D. Hirsch, Christos H. Papadimitriou, and Stephen A. Vavasis. “Exponential lower bounds for finding Brouwer fix points”. In: *J. Complex.* 5.4 (1989), pp. 379–416.
- [HR23] Alexandros Hollender and Aviad Rubinfeld. “Envy-Free Cake-Cutting for Four Agents”. In: *64th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2023, Santa Cruz, CA, USA, November 6-9, 2023*. IEEE, 2023, pp. 113–122. DOI: [10.1109/FOCS57990.2023.00015](https://doi.org/10.1109/FOCS57990.2023.00015). URL: <https://doi.org/10.1109/FOCS57990.2023.00015>.
- [KPR+09] Shiva Kintali, Laura J. Poplawski, Rajmohan Rajaraman, Ravi Sundaram, and Shang-Hua Teng. “Reducibility among Fractional Stability Problems”. In: *50th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2009, October 25-27, 2009, Atlanta, Georgia, USA*. IEEE Computer Society, 2009, pp. 283–292. DOI: [10.1109/FOCS.2009.57](https://doi.org/10.1109/FOCS.2009.57). URL: <https://doi.org/10.1109/FOCS.2009.57>.

- [OPR16] Abraham Othman, Christos H. Papadimitriou, and Aviad Rubinfeld. “The Complexity of Fairness Through Equilibrium”. In: *ACM Trans. Economics and Comput.* 4.4 (2016), 20:1–20:19. DOI: [10.1145/2956583](https://doi.org/10.1145/2956583). URL: <https://doi.org/10.1145/2956583>.
- [Pap94] Christos H. Papadimitriou. “On the Complexity of the Parity Argument and Other Inefficient Proofs of Existence”. In: *J. Comput. Syst. Sci.* 48.3 (1994), pp. 498–532.
- [Pro09] Ariel D. Procaccia. “Thou Shalt Covet Thy Neighbor’s Cake”. In: *IJCAI 2009, Proceedings of the 21st International Joint Conference on Artificial Intelligence, Pasadena, California, USA, July 11-17, 2009*. Ed. by Craig Boutilier. 2009, pp. 239–244. URL: <http://ijcai.org/Proceedings/09/Papers/049.pdf>.
- [Rub16] Aviad Rubinfeld. “Settling the Complexity of Computing Approximate Two-Player Nash Equilibria”. In: *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA*. Ed. by Irit Dinur. IEEE Computer Society, 2016, pp. 258–265. DOI: [10.1109/FOCS.2016.35](https://doi.org/10.1109/FOCS.2016.35). URL: <https://doi.org/10.1109/FOCS.2016.35>.
- [Rub18] Aviad Rubinfeld. “Inapproximability of Nash Equilibrium”. In: *SIAM J. Comput.* 47.3 (2018), pp. 917–959. DOI: [10.1137/15M1039274](https://doi.org/10.1137/15M1039274). URL: <https://doi.org/10.1137/15M1039274>.
- [Spe28] Emanuel Sperner. *Neuer, Beweis für die Invarianz der Dimensionszahl und des Gebietes*. 7. Vandenhoeck & Ruprecht, 1928.
- [Str08] Walter Stromquist. “Envy-Free Cake Divisions Cannot be Found by Finite Protocols”. In: *Electron. J. Comb.* 15.1 (2008). URL: [http://www.combinatorics.org/Volume%5C\\_15/Abstracts/v15i1r11.html](http://www.combinatorics.org/Volume%5C_15/Abstracts/v15i1r11.html).
- [Str80] Walter Stromquist. “How to cut a cake fairly”. In: *The American Mathematical Monthly* 87.8 (1980), pp. 640–644.
- [Su99] Francis Edward Su. “Rental Harmony: Sperner’s Lemma in Fair Division”. In: *The American Mathematical Monthly* 106.10 (1999), pp. 930–942. DOI: [10.1080/00029890.1999.12005142](https://doi.org/10.1080/00029890.1999.12005142).
- [VY11] Vijay V. Vazirani and Mihalis Yannakakis. “Market equilibrium under separable, piecewise-linear, concave utilities”. In: *J. ACM* 58.3 (2011), 10:1–10:25. DOI: [10.1145/1970392.1970394](https://doi.org/10.1145/1970392.1970394). URL: <https://doi.org/10.1145/1970392.1970394>.

## A Deferred Proofs

### A.1 Proof of Fact 28

**Fact 28.** Suppose  $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \mathbf{x}^{(3)}$  form a solution of  $C$ , then we have  $x_2^{(j)} \in [0.4, 0.6]$  and  $x_3^{(j)} \in [0.1, 0.3]$  for any  $j \in [3]$ .

Let  $\varepsilon = 2^{-n}$ ,  $\varepsilon_0 = 1.6\varepsilon$ . Let  $\mathcal{D} = \{(x, y, z) \in \Delta^2 : y \in 0.5 \pm 0.1, z \in 0.2 \pm 0.1\}$  denote the core region. We are going to show that  $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \mathbf{x}^{(3)} \in \mathcal{D}$ . We prove this fact by contradiction. W.l.o.g., we suppose that  $\mathbf{x}^{(1)}$  violates this condition. Since  $\|\mathbf{x}^{(i)} - \mathbf{x}^{(j)}\|_\infty \leq \varepsilon$ , all of them are inside the following subset:

$$\bar{\mathcal{D}} = \{y \in \Delta^2 : y_2 \notin [0.4 + \varepsilon, 0.6 - \varepsilon] \text{ or } y_3 \notin [0.1 + \varepsilon, 0.3 - \varepsilon]\} .$$

According to [Algorithm 1](#) and [Definition 23](#), we have

$$\forall \mathbf{y} \in \bar{D}, \quad C(\mathbf{y}) = \begin{cases} 1 & \text{if } y_2 \leq 0.5 \text{ and } y_3 \leq 0.1, \\ 2 & \text{if } y_2 > 0.5 \text{ and } y_3 \leq 0.1, \\ 1 & \text{if } y_2 \in [0.4, 0.4 + \varepsilon_0) \text{ and } y_3 \in [0.1, 0.1 + \varepsilon_0), \\ 2 & \text{if } y_2 \in [0.6 - \varepsilon_0, 0.6) \text{ and } y_3 \in [0.1, 0.1 + \varepsilon_0), \\ 1 \text{ or } 2 & \text{if } y_2 \in [0.4 + \varepsilon_0, 0.6 - \varepsilon_0) \text{ and } y_3 \in [0.1, 0.1 + \varepsilon], \\ 3 & \text{otherwise.} \end{cases}$$

Next, we discuss several cases on where  $\mathbf{x}^{(1)} \notin D$  is to establish the fact.

1. If  $x_2^{(1)} \leq 0.4$ , we have  $C(\mathbf{x}^{(1)}) \in \{1, 3\}$ . However, as any  $\mathbf{y} \in \bar{D}$  such that  $C(\mathbf{y}) = 2$  has  $y_2 \geq 0.4 + \varepsilon_0 > 0.4 + \varepsilon$ , the triangle formed by  $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \mathbf{x}^{(3)}$  cannot be trichromatic.
2. If  $x_2^{(1)} \geq 0.6$ , we have  $C(\mathbf{x}^{(1)}) \in \{2, 3\}$ . However, as any  $\mathbf{y} \in \bar{D}$  such that  $C(\mathbf{y}) = 1$  has  $y_2 < 0.6 - \varepsilon_0 < 0.6 - \varepsilon$ , the triangle formed by  $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \mathbf{x}^{(3)}$  cannot be trichromatic.
3. If  $x_3^{(1)} \leq 0.1$ , we have  $C(\mathbf{x}^{(1)}) \in \{1, 2\}$ . Note that any  $\mathbf{y} \in \bar{D}$  such that  $C(\mathbf{y}) = 3$  and  $y_3 \leq 0.1 + \varepsilon$  has  $y_2 \notin [0.4, 0.6]$ . If the triangle formed by  $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \mathbf{x}^{(3)}$  is trichromatic, we should further have  $x_2^{(1)}, x_2^{(2)}, x_2^{(3)} \leq 0.4 + \varepsilon$  or have  $x_2^{(1)}, x_2^{(2)}, x_2^{(3)} \geq 0.6 - \varepsilon$ . However, as discussed above, we cannot find  $C(\mathbf{x}^{(j)}) = 2$  with the first condition and we cannot find  $C(\mathbf{x}^{(j)}) = 1$  with the second condition. The triangle formed by  $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \mathbf{x}^{(3)}$  cannot be trichromatic.
4. If  $x_3^{(1)} \geq 0.3$ , we have  $C(\mathbf{x}^{(1)}) = 3$ . Note that there is no  $\mathbf{y} \in \bar{D}$  such that  $y_3 > 0.2$ . The triangle formed by  $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \mathbf{x}^{(3)}$  cannot be trichromatic.

## A.2 Proof of [Lemma 38](#)

**Lemma 38.** For any  $z, z' \in [0, 1]$  and any pair of points  $\mathbf{x}, \mathbf{x}'$  such that  $\mathbf{x} \in \{(0, 1 - z, z), (1 - z, 0, z)\}$  and  $\mathbf{x}' \in \{(0, 1 - z', z'), (1 - z', 0, z')\}$ , at least one of the following properties is satisfied:

- **Lipschitz in the hot&warm regions:**  $|\text{rel}^{nn}(\mathbf{x}) - \text{rel}^{nn}(\mathbf{x}')| \leq \varepsilon^{-2} \cdot |z - z'|$ ;
- **both are in the warm&cold regions:**  $\text{rel}^{nn}(\mathbf{x}), \text{rel}^{nn}(\mathbf{x}') \in \{0, 1\}$ .

According to the characterization [Lemma 34](#), for any  $z \in [0, 1]$ , we have

$$\text{rel}^{nn}(1 - z, 0, z), \text{rel}^{nn}(0, 1 - z, z) \in \begin{cases} \{0, 1\} & \text{if } z \leq 0.1 - 2\varepsilon^2; \\ \{0\} & \text{if } z \in (0.1 - 2\varepsilon^2, 0.1 - \varepsilon^2]; \\ \{0.5\varepsilon^{-2} \cdot (z - 0.1 + \varepsilon^2)\} & \text{if } z \in 0.1 \pm \varepsilon^2; \\ \{1\} & \text{if } z \in [0.1 + \varepsilon^2, 0.1 + 2\varepsilon^2); \\ \{0, 1\} & \text{if } z \geq 0.1 + 2\varepsilon^2. \end{cases}$$

That is, when  $z \in 0.1 \pm 2\varepsilon^2$ , we have an exact characterization for it. Note that the RHS of the characterization is continuous and  $0.5\varepsilon^{-2} \cdot (z - 0.1 + \varepsilon^2)$  is  $0.5\varepsilon^{-2}$ -Lipschitz. We have obtained this lemma for any  $z, z' \in 0.1 \pm 2\varepsilon^2$ .

On the other hand, w.l.o.g., suppose that  $z \notin 0.1 \pm 2\varepsilon^2$ . If  $|z - z'| \geq \varepsilon^2$ , this lemma is simply trivial because we have  $\varepsilon^{-2} \cdot |z - z'| \geq 1$ . Otherwise, according to the triangle inequality,  $z' \notin 0.1 \pm \varepsilon^2$ . Therefore, we have  $\text{rel}^{nn}(\mathbf{x}), \text{rel}^{nn}(\mathbf{x}') \in \{0, 1\}$ , which matches the second bullet of this lemma.

### A.3 Proof of Fact 55

**Fact 55.** Consider an auxillary function  $g(\mathbf{y}) = \alpha(y_3) \cdot (y_2 - 0.5)$ . For any  $\mathbf{y} \in \Delta^2$  such that  $y_3 \leq 0.06$ , warm and hot points satisfies  $g(\mathbf{y}) \in \pm 2\varepsilon^2$ . Furthermore,

$$d^\alpha(\mathbf{y}, \mathbf{nn}^\alpha(\mathbf{y})) = |g(\mathbf{y})|, \quad \text{and} \quad \text{rel}^{\mathbf{nn}, \alpha}(\mathbf{y}) = (0.5 + 0.5\varepsilon^{-2} \cdot g(\mathbf{y}))_{[0,1]}.$$

Note that in our base instance  $C$ , we have

$$\forall \mathbf{y}' \in \Delta^2, \quad \begin{cases} C(\mathbf{y}') = 1 & \text{if } y'_3 \leq 0.1, y'_2 \leq 0.5, \\ C(\mathbf{y}') = 2 & \text{if } y'_3 \leq 0.1, y'_2 > 0.5. \end{cases}$$

If  $y_2 \leq 0.5$ , we have  $C(\mathbf{y}) = 1$ .  $C(\mathbf{y}')$  is different with  $C(\mathbf{y})$  only if  $y'_2 > 0.5$  or  $y'_3 > 0.1$ . Since  $0.1 - y_3 > 0.04 \gg 2\varepsilon^2$ ,  $\mathbf{y}$  is hot or warm if and only if  $\alpha(y_3) \cdot (0.5 - y_2) < 2\varepsilon^2$ , i.e.,  $g(\mathbf{y}) \in (-2\varepsilon^2, 0)$ . In this case, we have  $d^\alpha(\mathbf{y}, \mathbf{nn}^\alpha(\mathbf{y})) = \alpha(y_3) \cdot |0.5 - y_2| = |g(\mathbf{y})|$ ,  $C_{\mathbf{nn}}^\alpha(\mathbf{y}) = 2$ , and

$$\text{rel}^{\mathbf{nn}, \alpha}(\mathbf{y}) = (0.5 - 0.5\varepsilon^{-2} \cdot |g(\mathbf{y})|)_+ = (0.5 + 0.5\varepsilon^{-2} \cdot g(\mathbf{y}))_{[0,1]}.$$

On the other hand, if  $y_2 > 0.5$ , we have  $C(\mathbf{y}) = 2$ .  $C(\mathbf{y}')$  is different with  $C(\mathbf{y})$  only if  $y'_2 \leq 0.5$  or  $y'_3 > 0.1$ . Since  $0.1 - y_3 > 0.04 \gg 2\varepsilon^2$ ,  $\mathbf{y}$  is hot or warm if and only if  $\alpha(y_3) \cdot (y_2 - 0.5) < 2\varepsilon^2$ , i.e.,  $g(\mathbf{y}) \in (0, 2\varepsilon^2)$ . In this case, we have  $d^\alpha(\mathbf{y}, \mathbf{nn}^\alpha(\mathbf{y})) = \alpha(y_3) \cdot |0.5 - y_2| = |g(\mathbf{y})|$ ,  $C_{\mathbf{nn}}^\alpha(\mathbf{y}) = 1$ , and

$$\text{rel}^{\mathbf{nn}, \alpha}(\mathbf{y}) = (0.5 + 0.5\varepsilon^{-2} \cdot |g(\mathbf{y})|)_- = (0.5 + 0.5\varepsilon^{-2} \cdot g(\mathbf{y}))_{[0,1]}.$$

### A.4 Proof of Lemma 65

**Lemma 65.** Given oracle access to a 2D-SPERNER instance  $C$  and a tuple of points  $(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \mathbf{x}^{(3)})$  which satisfy  $\|\mathbf{x}^{(i)} - \mathbf{x}^{(j)}\|_\infty \leq 2^{-4kn}$  for any  $i, j \in [3]$ , and which are trichromatic in the 3-out-of- $k+1$  Approximate Symmetric SPERNER instance  $C_{\text{sym}}^{(k)}$  constructed by Algorithm 4, then there is a polynomial-time algorithm that finds a tuple of points  $(\hat{\mathbf{x}}^{(1)}, \hat{\mathbf{x}}^{(2)}, \hat{\mathbf{x}}^{(3)})$  which satisfy  $\|\hat{\mathbf{x}}^{(i)} - \hat{\mathbf{x}}^{(j)}\|_\infty \leq 2^{-n}$  for any  $i, j \in [3]$  and which are trichromatic in the 2D-SPERNER instance  $C$ .

We assume that  $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \mathbf{x}^{(3)}$  give a trichromatic triangle in the 3-out-of- $k+1$  Approximate Symmetric SPERNER instance  $C_{\text{sym}}^{(k)}$  such that  $\|\mathbf{x}^{(i)} - \mathbf{x}^{(j)}\|_\infty \leq 2^{-4kn}$ , where  $k \geq 3$ . As in the earlier proof of Lemma 42, we use  $\mathbf{y}^{(i)}(\mathbf{x}) = (y_1^{(i)}(\mathbf{x}), y_2^{(i)}(\mathbf{x}), y_3^{(i)}(\mathbf{x}))$  to denote the *intermediate projections* and use  $\mathbf{c}^{(i)}(\mathbf{x}) = (c_1^{(i)}(\mathbf{x}), c_2^{(i)}(\mathbf{x}), c_3^{(i)}(\mathbf{x}))$  to denote the *intermediate palettes*. In addition, we introduce a new notation  $\tilde{y}_0(\mathbf{x})$  for the converted coordinate we compute on Line 3. In the rest of this section, because the subscripts we will use for  $\mathbf{c}^{(i)}$  can be very complicated, we will use  $c_j^{(i)}(\mathbf{x})$  and  $c^{(i)}(\mathbf{x}, j)$  interchangeably for better presentation. For any vector  $\mathbf{y} \in \Delta^2$ , we use  $i^*(\mathbf{y})$  to denote the first non-zero index of  $\mathbf{y}$ , i.e.,

$$i^*(\mathbf{y}) = \begin{cases} 1 & \text{if } y_1 > 0, \\ 2 & \text{if } y_1 = 0 \text{ and } y_2 > 0, \\ 3 & \text{otherwise.} \end{cases} \quad (24)$$

Our recovery algorithm (formalized in Algorithm 5) simulates Algorithm 4 for each  $\mathbf{x}^{(j)}$ . At time  $i \in [2, k-1]$  when we have computed  $\mathbf{y}^{(i)}(\mathbf{x}^{(j)})$  for each  $j \in [3]$ , we examine whether one of them is inside a trichromatic region, i.e., whether  $C$  has three different colors in  $\mathcal{N}(\mathbf{y}^{(i)}(\mathbf{x}^{(j)}))$  for some  $j \in [3]$ . According to our earlier discussions, this examination can be done in polynomial time. After we finish the simulation, we simply output  $\mathbf{y}^{(k)}(\mathbf{x}^{(1)}), \mathbf{y}^{(k)}(\mathbf{x}^{(2)}), \mathbf{y}^{(k)}(\mathbf{x}^{(3)})$  as a solution for  $C$ .

It is clear that any output during the simulation phase of this recovery algorithm gives a valid solution for the 2D-SPERNER instance  $C$ . To prove [Lemma 65](#), we only need to show that the three intermediate projections after the simulation form a valid solution for  $C$  if we output after the simulation phase. Or equivalently, if each  $\mathbf{y}^{(i)}(\mathbf{x}^{(j)})$  does not lie in a trichromatic region, the final converted coordinates,  $\mathbf{y}^{(k)}(\mathbf{x}^{(1)})$ ,  $\mathbf{y}^{(k)}(\mathbf{x}^{(2)})$ ,  $\mathbf{y}^{(k)}(\mathbf{x}^{(3)})$ , give a solution for  $C$ .

As in [Lemma 42](#), our proof will be considers two cases; the first is where the given solution  $\mathbf{x}^{(1)}$ ,  $\mathbf{x}^{(2)}$ ,  $\mathbf{x}^{(3)}$  for  $C_{\text{sym}}^{(k)}$  further enjoys the following property:

$$\forall 2 \leq i \leq k, \forall j \in [3], \quad P_{i+1}^{(k-i)}(\mathbf{x}^{(j)}) \leq 0.9. \quad (25)$$

One benefit of first considering this case is that we don't have too crazy projections in [Algorithm 5](#), which can help us significantly simplify the proof. The formal intermediate technical result is presented in [Corollary 86](#). Later, in the second step, we will explain how to prove [Lemma 65](#) in the case where this property is not satisfied.

**Case 1: the output satisfies Eq. (25).** We can continue to use the Lipschitzness of the projection step ([Lemma 43](#)) to obtain Lipschitzness for the intermediate projections.

**Lemma 84.** *Consider any  $k \geq 2$  and any  $\mathbf{x}^{(1)}, \mathbf{x}^{(2)} \in \Delta^k$ . Suppose that  $P_{i+1}^{(k-i)}(\mathbf{x}^{(j)}) \leq 0.9$  for any  $2 \leq i \leq k$  and any  $j \in [2]$ . Also, suppose that  $\mathcal{N}(\mathbf{y}^{(i)}(\mathbf{x}^{(j)}))$  is at most bichromatic for any  $2 \leq i < k$  and any  $j \in [2]$ . Then, for any  $2 \leq i \leq k$ , we have Lipschitzness for the third coordinate of the intermediate projections:*

$$|y_3^{(i)}(\mathbf{x}^{(1)}) - y_3^{(i)}(\mathbf{x}^{(2)})| \leq 2^{O(k)} \cdot \|\mathbf{x}^{(1)} - \mathbf{x}^{(2)}\|_\infty.$$

Furthermore, at least one of the following is satisfied for the second coordinates of the intermediate projections:

- **Lipschitz:**  $|y_2^{(i)}(\mathbf{x}^{(1)}) - y_2^{(i)}(\mathbf{x}^{(2)})| \leq 2^{3in+O(k)} \cdot \|\mathbf{x}^{(1)} - \mathbf{x}^{(2)}\|_\infty$ , or
- **both are on the left/right boundaries:** for any  $j \in [2]$ , we have either  $y_1^{(i)}(\mathbf{x}^{(j)}) = 0$  or  $y_2^{(i)}(\mathbf{x}^{(j)}) = 0$ .

*Proof.* We prove this lemma by induction on  $i$ . The base case is when  $i = 2$ . We have  $y_3^{(2)}(\mathbf{x}) = P_3^{(k-2)}(\mathbf{x})$  and thus we can obtain the Lipschitzness for  $y_3^{(2)}(\cdot)$  by

$$\begin{aligned} |y_3^{(i)}(\mathbf{x}^{(1)}) - y_3^{(i)}(\mathbf{x}^{(2)})| &\leq \|P^{(k-i)}(\mathbf{x}^{(1)}) - P^{(k-i)}(\mathbf{x}^{(2)})\|_\infty \\ &\leq 110 \cdot \|P^{(k-i-1)}(\mathbf{x}^{(1)}) - P^{(k-i-1)}(\mathbf{x}^{(2)})\|_\infty \\ &\leq \dots \\ &\leq 110^{k-i} \cdot \|\mathbf{x}^{(1)} - \mathbf{x}^{(2)}\|_\infty \leq 2^{7k} \cdot \|\mathbf{x}^{(1)} - \mathbf{x}^{(2)}\|_\infty, \end{aligned}$$

which works for any  $i \geq 2$ . At the same time, we can similarly obtain  $|P_2^{(k-1)}(\mathbf{x}^{(1)}) - P_2^{(k-1)}(\mathbf{x}^{(2)})| \leq 2^{7k} \cdot \|\mathbf{x}^{(1)} - \mathbf{x}^{(2)}\|_\infty$ . Because the function  $g(y_0) = (0.5 + 0.5\epsilon^{-2} \cdot (y_0 - 0.1))_{[0,1]}$  is clearly  $0.5\epsilon^{-2}$ -Lipschitz, we have

$$|\tilde{y}_0(\mathbf{x}^{(1)}) - \tilde{y}_0(\mathbf{x}^{(2)})| \leq 2^{2n} \cdot |P_2^{(k-1)}(\mathbf{x}^{(1)}) - P_2^{(k-1)}(\mathbf{x}^{(2)})| \leq 2^{2n+7k} \cdot \|\mathbf{x}^{(1)} - \mathbf{x}^{(2)}\|_\infty.$$

Hence, we can the first bullet of the second statement of this lemma.

$$|y_2^{(2)}(\mathbf{x}^{(1)}) - y_2^{(2)}(\mathbf{x}^{(2)})| \leq |\tilde{y}_0(\mathbf{x}^{(1)}) - \tilde{y}_0(\mathbf{x}^{(2)})| + |y_3^{(2)}(\mathbf{x}^{(1)}) - y_3^{(2)}(\mathbf{x}^{(2)})| \leq 2^{2n+7k+1} \cdot \|\mathbf{x}^{(1)} - \mathbf{x}^{(2)}\|_\infty.$$



Consider any  $i_0 \geq 3$ . Suppose that we have proved this lemma for  $i < i_0$ . Next, we consider when  $i = i_0$ . Note that we have  $y_3^{(i)}(\mathbf{x}) = P_{i+1}^{(k-i)}(\mathbf{x})$  for any  $\mathbf{x}$ . By the premise that  $P_{i+1}^{(k-i)}(\mathbf{x}^{(j)}) \leq 0.9$ , we can use [Lemma 43](#) to get the first statement of this lemma by the earlier inequalities.

Next, we establish the second statement of this lemma, in which we need to prove at least one of the bullets is satisfied. For the induction hypothesis, we will use the following more specific version of the first bullet

$$|y_2^{(i)}(\mathbf{x}^{(1)}) - y_2^{(i)}(\mathbf{x}^{(2)})| \leq 2^{3in+7k+\log 2i} \cdot \|\mathbf{x}^{(1)} - \mathbf{x}^{(2)}\|_\infty.$$

Note that  $y_2^{(i)}(\mathbf{x}) = (1 - y_3^{(i)}(\mathbf{x})) \cdot \text{rel}^{nn,\alpha}(\mathbf{y}^{(i-1)}(\mathbf{x}))$ . Then, it is easy to obtain that

$$\begin{aligned} \left| y_2^{(i)}(\mathbf{x}^{(1)}) - y_2^{(i)}(\mathbf{x}^{(2)}) \right| &= \left| (1 - y_3^{(i)}(\mathbf{x}^{(1)})) \cdot \text{rel}^{nn,\alpha}(\mathbf{y}^{(i-1)}(\mathbf{x}^{(1)})) - (1 - y_3^{(i)}(\mathbf{x}^{(2)})) \cdot \text{rel}^{nn,\alpha}(\mathbf{y}^{(i-1)}(\mathbf{x}^{(2)})) \right| \\ &\leq \text{rel}^{nn,\alpha}(\mathbf{y}^{(i-1)}(\mathbf{x}^{(1)})) \cdot \left| y_3^{(i)}(\mathbf{x}^{(1)}) - y_3^{(i)}(\mathbf{x}^{(2)}) \right| \\ &\quad + (1 - y_3^{(i)}(\mathbf{x}^{(2)})) \cdot \left| \text{rel}^{nn,\alpha}(\mathbf{y}^{(i-1)}(\mathbf{x}^{(1)})) - \text{rel}^{nn,\alpha}(\mathbf{y}^{(i-1)}(\mathbf{x}^{(2)})) \right| \\ &\leq \left| y_3^{(i)}(\mathbf{x}^{(1)}) - y_3^{(i)}(\mathbf{x}^{(2)}) \right| + \left| \text{rel}^{nn,\alpha}(\mathbf{y}^{(i-1)}(\mathbf{x}^{(1)})) - \text{rel}^{nn,\alpha}(\mathbf{y}^{(i-1)}(\mathbf{x}^{(2)})) \right| \\ &\leq 2^{7k} \cdot \|\mathbf{x}^{(1)} - \mathbf{x}^{(2)}\|_\infty + \left| \text{rel}^{nn,\alpha}(\mathbf{y}^{(i-1)}(\mathbf{x}^{(1)})) - \text{rel}^{nn,\alpha}(\mathbf{y}^{(i-1)}(\mathbf{x}^{(2)})) \right|. \end{aligned}$$

Suppose that our induction hypothesis gives the first bullet for  $i - 1$ . Combining the Lipschitzness on the third coordinate, we have

$$\|\mathbf{y}^{(i-1)}(\mathbf{x}^{(1)}) - \mathbf{y}^{(i-1)}(\mathbf{x}^{(2)})\|_\infty \leq (2^{3(i-1)n+7k+\log 2(i-1)} + 2^{7k}) \cdot \|\mathbf{x}^{(1)} - \mathbf{x}^{(2)}\|_\infty.$$

If  $\text{rel}^{nn,\alpha}(\mathbf{y}^{(i-1)}(\mathbf{x}^{(1)})), \text{rel}^{nn,\alpha}(\mathbf{y}^{(i-1)}(\mathbf{x}^{(2)})) \in \{0, 1\}$ , we have the second bullet for  $i = i_0$ . Otherwise, according to [Lemma 54](#), we have

$$\begin{aligned} \left| y_2^{(i)}(\mathbf{x}^{(1)}) - y_2^{(i)}(\mathbf{x}^{(2)}) \right| &\leq 2^{7k} \cdot \|\mathbf{x}^{(1)} - \mathbf{x}^{(2)}\|_\infty + 2^{3n} \cdot \left| \mathbf{y}^{(i-1)}(\mathbf{x}^{(1)}) - \mathbf{y}^{(i-1)}(\mathbf{x}^{(2)}) \right| \\ &\leq 2^{7k} \cdot \|\mathbf{x}^{(1)} - \mathbf{x}^{(2)}\|_\infty + 2^{3n} \cdot (2^{3(i-1)n+7k+\log 2(i-1)} + 2^{7k}) \cdot \|\mathbf{x}^{(1)} - \mathbf{x}^{(2)}\|_\infty \\ &\leq 2^{7k} \cdot \|\mathbf{x}^{(1)} - \mathbf{x}^{(2)}\|_\infty + (2i - 1) \cdot 2^{3in+7k} \cdot \|\mathbf{x}^{(1)} - \mathbf{x}^{(2)}\|_\infty \\ &\leq 2^{3in+7k+\log 2i} \cdot \|\mathbf{x}^{(1)} - \mathbf{x}^{(2)}\|_\infty, \end{aligned}$$

which gives the first bullet for  $i = i_0$ .

On the other hand, suppose that the induction hypothesis further gives us the second bullet for  $i - 1$ . That is, for any  $j \in [2]$ , we have

$$\mathbf{y}^{(i-1)}(\mathbf{x}^{(j)}) \in \left\{ \left( 0, 1 - y_3^{(i-1)}(\mathbf{x}^{(j)}), y_3^{(i-1)}(\mathbf{x}^{(j)}) \right), \left( 1 - y_3^{(i-1)}(\mathbf{x}^{(j)}), 0, y_3^{(i-1)}(\mathbf{x}^{(j)}) \right) \right\}.$$

If  $\text{rel}^{nn,\alpha}(\mathbf{y}^{(i-1)}(\mathbf{x}^{(1)})), \text{rel}^{nn,\alpha}(\mathbf{y}^{(i-1)}(\mathbf{x}^{(2)})) \in \{0, 1\}$ , we have the second bullet for  $i = i_0$ . Otherwise, according to [Lemma 56](#), we have

$$\begin{aligned} \left| y_2^{(i)}(\mathbf{x}^{(1)}) - y_2^{(i)}(\mathbf{x}^{(2)}) \right| &\leq 2^{7k} \cdot \|\mathbf{x}^{(1)} - \mathbf{x}^{(2)}\|_\infty + 2^{2n} \cdot \left| y_3^{(i-1)}(\mathbf{x}^{(1)}) - y_3^{(i-1)}(\mathbf{x}^{(2)}) \right| \\ &\leq 2^{7k} \cdot \|\mathbf{x}^{(1)} - \mathbf{x}^{(2)}\|_\infty + 2^{2n+7k} \cdot \|\mathbf{x}^{(1)} - \mathbf{x}^{(2)}\|_\infty \\ &\leq 2^{2in+7k+1} \cdot \|\mathbf{x}^{(1)} - \mathbf{x}^{(2)}\|_\infty, \end{aligned}$$

which gives the first bullet for  $i = i_0$ . □

Suppose that [Algorithm 5](#) fails to give us any solution during the simulation phase, i.e., for any  $2 \leq i \leq k-1$  and  $j \in [3]$ ,  $\mathcal{N}(\mathbf{y}^{(i)}(\mathbf{x}^{(j)}))$  is at most bichromatic. Since the solution of the 3-out-of- $k+1$  Approximate Symmetric SPERNER instance satisfies  $\|\mathbf{x}^{(j_1)} - \mathbf{x}^{(j_2)}\|_\infty \leq 2^{-4kn}$ , we have  $|y_3^{(i)}(\mathbf{x}^{(j_1)}) - y_3^{(i)}(\mathbf{x}^{(j_2)})| < 2^{-5n} = \varepsilon^5$  for any  $2 \leq i \leq k$  and any  $j_1, j_2 \in [3]$ , and further that

- **the intermediate projections are close to each other:** for any  $j_1, j_2 \in [3]$ ,  $|y_2^{(k)}(\mathbf{x}^{(j_1)}) - y_2^{(k)}(\mathbf{x}^{(j_2)})| < 2^{-2n} = \varepsilon^2$  and  $|y_2^{(i)}(\mathbf{x}^{(j_1)}) - y_2^{(i)}(\mathbf{x}^{(j_2)})| < 2^{-5n} = \varepsilon^5$  for  $i < k$ ; or
- **all intermediate projections are on the left/right boundaries:** for any  $j \in [3]$ , we have either  $y_1^{(i)}(\mathbf{x}^{(j)}) = 0$  or  $y_2^{(i)}(\mathbf{x}^{(j)}) = 0$ .

Next, we prove the same characterization of the intermediate palettes used in [Algorithm 4](#) by [Lemma 85](#). The characterization gives equivalence between the set of relevant colors in the palette. In the first case, where the intermediate projections of the three input vectors are close to each other (and at least one of them is not on the left/right boundaries), the palettes are exactly the same. In the second case, where the intermediate projections of the input vectors are all on the left/right boundaries, the only two relevant colors for the points,  $c^{(i)}(\mathbf{x}, i^*(\mathbf{y}^{(i)}(\mathbf{x})))$  and  $c_3^{(i)}(\mathbf{x})$ , are respectively equal.

**Lemma 85.** *Suppose that  $P_{i+1}^{(k-i)}(\mathbf{x}^{(j)}) \leq 0.9$  for any  $2 \leq i \leq k$  and any  $j \in [3]$ . Suppose that  $\mathcal{N}(\mathbf{y}^{(i)}(\mathbf{x}^{(j)}))$  is not trichromatic for any  $2 \leq i < k$  and  $j \in [3]$ . For any  $2 \leq i \leq k$ , at least one of the following holds:*

- *The intermediate projections are close to each other, and we have that the corresponding palettes are the same:  $c^{(i)}(\mathbf{x}^{(j_1)}) = c^{(i)}(\mathbf{x}^{(j_2)})$  for any  $j_1, j_2 \in [3]$ .*
- *All intermediate projections are on the left/right boundaries, and the palettes may be different on an irrelevant color, but we still have that both of the following hold:*
  - *The color of the first non-zero coordinate  $c^{(i)}(\mathbf{x}^{(j)}, i^*(\mathbf{y}^{(i)}(\mathbf{x}^{(j)})))$  is the same across all  $j \in [3]$ ; and*
  - *the 3rd color is the same,  $c_3^{(i)}(\mathbf{x}^{(j)}) = i + 1$  for any  $j \in [3]$ .*

Since the output  $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \mathbf{x}^{(3)}$  is trichromatic in  $C_{\text{sym}}^{(k)}$ , we have

$$\left| \{c^{(k)}(\mathbf{x}^{(j)}, C(\mathbf{y}^{(k)}(\mathbf{x}^{(j)})))\}_{j \in [3]} \right| = 3. \quad (26)$$

We should always have the first bullet of [Lemma 85](#) for  $k$  when each  $\mathcal{N}(\mathbf{y}^{(i)}(\mathbf{x}^{(j)}))$  is not trichromatic, because otherwise the second bullet of [Lemma 85](#) violates [Eq. \(26\)](#) as  $C(\mathbf{y}^{(k)}(\mathbf{x}^{(j)})) \in \{i^*(\mathbf{y}^{(k)}(\mathbf{x}^{(j)})), 3\}$ . Note that the first bullet of [Lemma 45](#) and [Eq. \(10\)](#) imply that the colors in the base instance  $C(\mathbf{y}^{(k)}(\mathbf{x}^{(1)}))$ ,  $C(\mathbf{y}^{(k)}(\mathbf{x}^{(2)}))$ , and  $C(\mathbf{y}^{(k)}(\mathbf{x}^{(3)}))$  should be distinct. We can always guarantee that the tuple  $\mathbf{y}^{(k)}(\mathbf{x}^{(1)}), \mathbf{y}^{(k)}(\mathbf{x}^{(2)}), \mathbf{y}^{(k)}(\mathbf{x}^{(3)})$  gives a solution to  $C$ .

**Corollary 86.** *Suppose that  $P_{i+1}^{(k-i)}(\mathbf{x}^{(j)}) \leq 0.9$  for any  $2 \leq i \leq k$  and any  $j \in [3]$ . Suppose that  $\mathcal{N}(\mathbf{y}^{(i)}(\mathbf{x}^{(j)}))$  is not trichromatic for any  $2 \leq i < k$  and  $j \in [3]$ . Then, we have  $|\{C(\mathbf{y}^{(k)}(\mathbf{x}^{(j)})) : j \in [3]\}| = 3$ .*

Before giving the proof of [Lemma 85](#), we give an useful lemma here. The lemma states that any point very close (e.g.,  $< \varepsilon^3$ ) to a hot point should be either hot or warm. This could resolve the problem arisen from the fact that  $d^\alpha(\cdot, \cdot)$  is only quasimetric and does not enjoy the triangle inequality.

**Lemma 87.** *For any  $\mathbf{x}$  that is hot and any  $\mathbf{x}'$  such that  $d(\mathbf{x}, \mathbf{x}') \leq \varepsilon^3$ ,  $\mathbf{x}'$  is either hot or warm.*

*Proof.* If  $\mathbf{x}, \mathbf{x}'$  has different colors, both of them are hot. Next, we consider that  $\mathbf{x}, \mathbf{x}'$  have the same color. Let  $\mathbf{y} = \mathbf{nn}^\alpha(\mathbf{x})$  denote the nearest neighbor of  $\mathbf{x}$ . If  $x_3, x'_3 \geq 0.05$ , because of [Observation 50](#) and the fact  $d(\cdot, \cdot)$  satisfies the triangle inequality, the lemma holds as

$$d^\alpha(\mathbf{x}', \mathbf{y}) = d(\mathbf{x}', \mathbf{y}) \leq d(\mathbf{x}, \mathbf{x}') + d(\mathbf{x}, \mathbf{y}) = d(\mathbf{x}, \mathbf{x}') + d^\alpha(\mathbf{x}, \mathbf{y}) < \varepsilon^3 + \varepsilon^2 < 2\varepsilon^2.$$

Further, we consider that  $x_3, x'_3 \leq 0.06$  to finish the proof (because we have  $|x_3 - x'_3| \leq \varepsilon^3$ ). Since  $\mathbf{x}$  is hot, we have  $d^\alpha(\mathbf{x}, \mathbf{y}) < \varepsilon^2$ , and thus

$$\alpha(x_3) \cdot |x_2 - y_2|, |x_3 - y_3| < \varepsilon^2.$$

Because  $d(\mathbf{x}, \mathbf{x}') \leq \varepsilon^3$ , we have  $|x_2 - x'_2|, |x_3 - x'_3| \leq \varepsilon^3$ , and thus

$$\begin{aligned} \alpha(x'_3) \cdot |x'_2 - y_2| &\leq |\alpha(x_3) - \alpha(x'_3)| + \alpha(x_3) \cdot |x'_2 - y_2| \\ &\leq O(n) \cdot |x_3 - x'_3| + \alpha(x_3) \cdot (|x'_2 - x_2| + |x_2 - y_2|) && \text{(Lemma 49)} \\ &\leq O(n) \cdot \varepsilon^3 + \varepsilon^3 + \varepsilon^2 < 2\varepsilon^2, \\ |x'_3 - y_3| &\leq |x_3 - x'_3| + |x_3 - y_3| < \varepsilon^3 + \varepsilon^2 < 2\varepsilon^2. \end{aligned}$$

We have  $d^\alpha(\mathbf{x}', \mathbf{y}) < 2\varepsilon^2$ . Because  $\mathbf{y}$  (or, an infinite sequence of points that converges to  $\mathbf{y}$ ) has a different color with  $\mathbf{x}, \mathbf{x}', \mathbf{x}'$  is hot or warm.  $\square$

*Proof of Lemma 85.* We prove this lemma by induction. The base case is when  $i = 2$ . At the beginning of [Algorithm 4](#), we have  $\|\mathbf{y}^{(2)}(\mathbf{x}^{(j_1)}) - \mathbf{y}^{(2)}(\mathbf{x}^{(j_2)})\|_\infty \leq 2^{O(k)} \cdot 2^{-4kn} < \varepsilon^5$  for any  $j_1, j_2 \in [3]$  according to the proof of [Lemma 84](#) and  $\mathbf{c}^{(2)}(\mathbf{x}^{(j)}) = (1, 2, 3)$  for any  $j \in [3]$ . We have the first bullet satisfied for  $i = 2$ .

Consider any  $i_0 \geq 3$ . Assume that we have established this lemma for any  $i = i_0 - 1$ . Next, we establish this lemma for  $i = i_0$ .

First, consider that the first bullet holds for  $i - 1$ . We have  $|y_2^{(i-1)}(\mathbf{x}^{(j_1)}) - y_2^{(i-1)}(\mathbf{x}^{(j_2)})| < 2^{-5n} = \varepsilon^5$  for any  $j_1, j_2 \in [3]$ . Hence,  $d(\mathbf{y}^{(i-1)}(\mathbf{x}^{(j_1)}), \mathbf{y}^{(i-1)}(\mathbf{x}^{(j_2)})) \leq \varepsilon^5$  for any  $j_1, j_2 \in [3]$ . We discuss two cases on whether there is a hot point in  $\mathbf{y}^{(i-1)}(\mathbf{x}^{(1)})$ ,  $\mathbf{y}^{(i-1)}(\mathbf{x}^{(2)})$  and  $\mathbf{y}^{(i-1)}(\mathbf{x}^{(3)})$ .

- W.l.o.g., suppose that  $\mathbf{y}^{(i-1)}(\mathbf{x}^{(1)})$  is a hot point. According to [Lemma 54](#) and [Lines 4 to 6](#), all the  $i$ -th intermediate projections are close to each other. According to [Lemma 87](#),  $\mathbf{y}^{(i-1)}(\mathbf{x}^{(2)})$  and  $\mathbf{y}^{(i-1)}(\mathbf{x}^{(3)})$  are also hot or warm. Since they are all in a region that is at most bichromatic, the following color set  $\{C(\mathbf{y}^{(i-1)}(\mathbf{x}^{(j)})), \hat{C}_{\text{nn}}^\alpha(\mathbf{y}^{(i-1)}(\mathbf{x}^{(j)}))\}$  is then the same across all  $j \in [3]$ . Since  $\mathbf{c}^{(i-1)}(\mathbf{x}^{(1)}) = \mathbf{c}^{(i-1)}(\mathbf{x}^{(2)}) = \mathbf{c}^{(i-1)}(\mathbf{x}^{(3)})$ , we then have the same  $\mathbf{c}^{(i)}(\mathbf{x}^{(j)})$  across all  $j \in [3]$ , which gives the first bullet of this lemma.
- Otherwise, suppose that  $\mathbf{y}^{(i-1)}(\mathbf{x}^{(j)})$  is warm or cold for each  $j \in [3]$ . We have  $\text{rel}^{\text{nn}, \alpha}(\mathbf{y}^{(i-1)}(\mathbf{x}^{(j)})) \in \{0, 1\}$  for any  $j \in [3]$  in this case. The  $i$ -th intermediate projections are on the left/right boundaries, i.e., we have  $y_1^{(i)}(\mathbf{x}^{(j)}) = 0$  or  $y_2^{(i)}(\mathbf{x}^{(j)}) = 0$  for each  $j \in [3]$ . Note that there is only one color in  $\{C(\mathbf{y}^{(i-1)}(\mathbf{x}^{(j)}))\}_{j \in [3]}$ , because otherwise we clearly have  $d^\alpha(\mathbf{y}^{(i-1)}(\mathbf{x}^{(j)}), \mathbf{nn}^\alpha(\mathbf{y}^{(i-1)}(\mathbf{x}^{(j)}))) < \varepsilon^2$  and all points are hot. Note that [Algorithm 4](#) ensures in this scenario that

$$\begin{aligned} i^*(\mathbf{y}^{(i)}(\mathbf{x}^{(j)})) = 1 &\Leftrightarrow \text{rel}^{\text{nn}, \alpha}(\mathbf{y}^{(i-1)}(\mathbf{x}^{(j)})) = 0 \\ &\Leftrightarrow C(\mathbf{y}^{(i-1)}(\mathbf{x}^{(j)})) < \hat{C}_{\text{nn}}^\alpha(\mathbf{y}^{(i-1)}(\mathbf{x}^{(j)})) \\ &\Leftrightarrow c^{(i)}(\mathbf{x}^{(j)}, 1) = c^{(i-1)}(\mathbf{x}^{(j)}, C(\mathbf{y}^{(i-1)}(\mathbf{x}^{(j)}))) \end{aligned}$$

where the second *iff* is obtained by [Fact 60](#). We have  $c^{(i)}(\mathbf{x}^{(j)}, i^*(\mathbf{y}^{(i)}(\mathbf{x}^{(j)}))) = c^{(i-1)}(\mathbf{x}^{(j)}, C(\mathbf{y}^{(i-1)}(\mathbf{x}^{(j)})))$  for each  $j \in [3]$ . Since  $\mathbf{c}^{(i-1)}(\mathbf{x}^{(1)}) = \mathbf{c}^{(i-1)}(\mathbf{x}^{(2)}) = \mathbf{c}^{(i-1)}(\mathbf{x}^{(3)})$  and  $C(\mathbf{y}^{(i-1)}(\mathbf{x}^{(1)})) = C(\mathbf{y}^{(i-1)}(\mathbf{x}^{(2)})) = C(\mathbf{y}^{(i-1)}(\mathbf{x}^{(3)}))$ , we have the second bullet for  $i$ .

Second, consider that the second bullet holds for  $i - 1$ . Because  $|y_3^{(i-1)}(\mathbf{x}^{(j_1)}) - y_3^{(i-1)}(\mathbf{x}^{(j_2)})| < \varepsilon^5$  for any  $j_1, j_2 \in [3]$ , according to the characterization of the temperature on the left/right boundaries (Lemma 53), we have

- for each  $j \in [3]$ ,  $\mathbf{y}^{(i-1)}(\mathbf{x}^{(j)})$  is hot or warm; or
- for each  $j \in [3]$ ,  $\mathbf{y}^{(i-1)}(\mathbf{x}^{(j)})$  is warm or cold.

Since we have  $C(1 - z, 0, z) = 3 = C(0, 1 - z, z)$  or  $C(1 - z, 0, z) = 1, C(0, 1 - z, z) = 2$  for any  $z \in [0, 1]$  (Fact 26), we have the same  $c^{(i-1)}(\mathbf{x}^{(j)}, C(\mathbf{y}^{(i-1)}(\mathbf{x}^{(j)})))$  across all  $j \in [3]$ . Next, we discuss the above two cases to finish the proof.

- Consider when each  $\mathbf{y}^{(i-1)}(\mathbf{x}^{(j)})$  is hot or warm. We have  $\hat{C}_{\text{nn}}^\alpha(\mathbf{y}^{(i-1)}(\mathbf{x}^{(j)})) = C_{\text{nn}}^\alpha(\mathbf{y}^{(i-1)}(\mathbf{x}^{(j)}))$  for each  $j \in [3]$ . Because we have  $C_{\text{nn}}^\alpha(1 - z, 0, z) = 1, C_{\text{nn}}^\alpha(0, 1 - z, z) = 2$ , or  $C_{\text{nn}}^\alpha(1 - z, 0, z) = 3 = C_{\text{nn}}^\alpha(0, 1 - z, z)$  for each  $z \in 0.1 \pm 2\varepsilon^2$  (Lemma 53), we have the same  $c^{(i-1)}(\mathbf{x}^{(j)}, \hat{C}_{\text{nn}}^\alpha(\mathbf{y}^{(i-1)}(\mathbf{x}^{(j)})))$  across all  $j \in [3]$ . Therefore, the palette  $c^{(i)}(\mathbf{x}^{(j)})$  is the same across all  $j \in [3]$ . Because of Lemma 84 and the Lipschitzness of the coordinate converter on the left/right boundaries Lemma 56, the  $i$ -th intermediate projections,  $\mathbf{y}^{(i)}(\mathbf{x}^{(1)}), \mathbf{y}^{(i)}(\mathbf{x}^{(2)})$  and  $\mathbf{y}^{(i)}(\mathbf{x}^{(3)})$ , are close to each other. Hence, we prove the first bullet of this lemma for  $i$ .
- Consider when each  $\mathbf{y}^{(i-1)}(\mathbf{x}^{(j)})$  is warm or cold. We have  $\text{rel}^{\text{nn}, \alpha}(\mathbf{y}^{(i-1)}(\mathbf{x}^{(j)})) \in \{0, 1\}$  and all  $i$ -th intermediate projections are on the left/right boundaries. Note that there is only one color in  $\{C(\mathbf{y}^{(i-1)}(\mathbf{x}^{(j)}))\}_{j \in [3]}$ , because otherwise we clearly have  $d^\alpha(\mathbf{y}^{(i-1)}(\mathbf{x}^{(j)}), \text{nn}(\mathbf{y}^{(i-1)}(\mathbf{x}^{(j)}))) < \varepsilon^2$  and all points are hot. According to our earlier discussions, Algorithm 4 ensures in this scenario that  $c^{(i)}(\mathbf{x}^{(j)}, i^*(\mathbf{y}^{(i)}(\mathbf{x}^{(j)}))) = c^{(i-1)}(\mathbf{x}^{(j)}, C(\mathbf{y}^{(i-1)}(\mathbf{x}^{(j)})))$ . Therefore, we have the same  $c^{(i)}(\mathbf{x}^{(j)}, i^*(\mathbf{y}^{(i)}(\mathbf{x}^{(j)})))$  across all  $j \in [3]$ , and thus the second bullet of this lemma holds.  $\square$

**Case 2: the output does not satisfy Eq. (25).** Next, we complete our second step by showing how to prove Lemma 65 without the property (Eq. (25)). Suppose  $\theta^*$  is the minimum threshold  $\theta \geq 2$  such that  $P_{i+1}^{(k-i)}(\mathbf{x}^{(j)}) \leq 0.9$  for any  $\theta \leq i \leq k$  and  $j \in [3]$ . Such threshold always exists because otherwise we have  $x_{k+1}^{(1)}, x_{k+1}^{(2)}, x_{k+1}^{(3)} \geq 0.8$ . This implies  $y_3^{(k)}(\mathbf{x}^{(j)}) > 0.8$  for any  $j \in [3]$ , and according to our construction of the base instance (Algorithm 1), we have  $C^{(k)}(\mathbf{x}^{(j)}) = c_3^{(k)}(\mathbf{x}^{(j)}) = k + 1$  for any  $j \in [3]$ , violating the assumption that  $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \mathbf{x}^{(3)}$  form a solution for the 3-out-of- $k+1$  Approximate Symmetric SPERNER problem. When  $\theta^* = 2$ , it is equivalent with the special case satisfying Eq. (25) and we have proved Lemma 65 for this case. On the other hand, if  $\theta^* > 2$ , we have  $P_{\theta^*}^{(k-\theta^*+1)}(\mathbf{x}^{(j)}) \geq 0.8$  for any  $j \in [3]$  because of the Lipschitzness of the projection step Lemma 43 and that  $\|\mathbf{x}^{(i)} - \mathbf{x}^{(j)}\|_\infty \leq 2^{-4kn}$  for any  $i, j \in [3]$ . This means that we have  $C(\mathbf{y}^{(\theta^*-1)}(\mathbf{x}^{(j)})) = 3$ . And because of Fact 57, we have  $\text{rel}^{\text{nn}, \alpha}(\mathbf{y}^{(\theta^*-1)}(\mathbf{x}^{(j)})) = 1$  and  $y_1^{(\theta^*)}(\mathbf{x}^{(j)}) = 0$  for any  $j \in [3]$ . Further, we have for each  $j \in [3]$ ,

$$\mathbf{y}^{(\theta^*)}(\mathbf{x}^{(j)}) = \left( 0, 1 - P_{\theta^*+1}^{(k-\theta^*)}(\mathbf{x}^{(j)}), P_{\theta^*+1}^{(k-\theta^*)}(\mathbf{x}^{(j)}) \right). \quad (27)$$

Therefore, running Algorithm 4 on instance  $C^{(k)}$  for  $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \mathbf{x}^{(3)}$  is equivalent to running Algorithm 4 on instance  $C^{(k')}$  with  $k' = k - \theta^* + 2$  for  $\hat{\mathbf{x}}^{(1)}, \hat{\mathbf{x}}^{(2)}, \hat{\mathbf{x}}^{(3)} \in \Delta^{k'}$  such that

$$\forall i \in [k' + 1], j \in [3], \quad \hat{x}_i^{(j)} = \begin{cases} 0 & \text{if } i = 1, \\ 1 - \sum_{i'=\theta^*+1}^{k+1} x_{i'}^{(j)} & \text{if } i = 2, \\ x_{i+\theta^*-2}^{(j)} & \text{if } i > 2. \end{cases}$$

This is because  $\mathbf{y}^{(2)}(\hat{\mathbf{x}}^{(j)})$  equals the RHS of Eq. (27). Because this new instance has a smaller number of dimensions and satisfies the condition of our special cases (Eq. (25)), we complete the proof for Lemma 65.