Improved Hardness of Approximating *k-Clique* under *ETH*

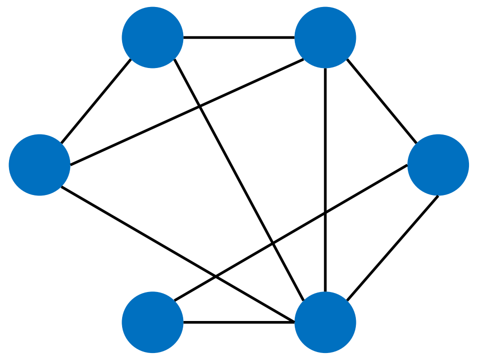
Bingkai Lin, *Xuandi Ren*, Yican Sun, Xiuhan Wang

Nanjing U, *UC Berkeley*, Peking U, Tsinghua U

November 2023

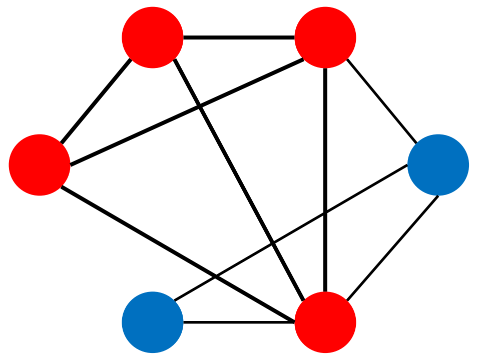
k-Clique Problem

Input: an undirected graph $G = (V, E)$, an integer k.
Output: whether there is a clique of size k in G.



k-Clique Problem

Input: an undirected graph $G = (V, E)$, an integer k.
Output: whether there is a clique of size k in G.



- Let $n = |V|$, then k-Clique problem is
  - **NP-complete** [Karp'72]
    - does not admit $n^{O(1)}$ time algorithm assuming NP≠P
  - **W[1]-complete** [Downey-Fellows'95]
    - does not admit $f(k) \cdot n^{O(1)}$ time algorithm assuming W[1]≠FPT

Hardness of Approximating k-Clique

- A c-approximation algorithm for k-Clique can:
  - find a clique of size k/c whenever there is a clique of size k in G.
  - (equivalently) distinguish between: G has a k-clique, or G has no (k/c)-clique.

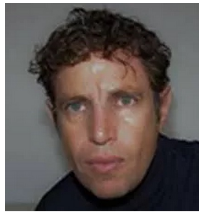
Approximating Clique is Almost NP-Complete. **FOCS** 1991

Test of Time Award
FOCS 2021

Interactive Proofs and the Hardness of Approximating Cliques. **JACM** 1996

Gödel Prize
2001




Uriel **F**eige, Shafi **G**oldwasser, László **L**ovász, Shmuel **S**afra, Mario **S**zegedy

- The first polynomial time inapproximability of k-Clique
- Motivated the discovery of the PCP theorem

Hardness of Approximating k-Clique

- After a long line of work [BGLR93, BS94, FGLSS96, Has96, BGS97, Gol98, FK00, Zuc07]:

$n^{1-\epsilon}$-approximating k-Clique is NP-hard

- New research problem from parameterized complexity

Does k-Clique have $f(k) \cdot n^{o(k/\gamma(k))}$ time $\gamma(k)$-approximation algorithm?

Hardness of Approximating k-Clique

- New research problem from parameterized complexity

Does k-Clique have $f(k) \cdot n^{o(k/\gamma(k))}$ time $\gamma(k)$-approximation algorithm?

- Polynomial time inapproximability does not rule out $n^{o(k/\gamma(k))}$ time algorithm.
- Assuming Gap-ETH, the answer is NO. [CCK+'17]
- It is more interesting to prove inapproximability under ETH:

From Gap-k-Clique to PIH [LRSW'22]

An $\boldsymbol{f(k) \cdot n^{\omega\left(\frac{k}{\log k}\right)}}$-time lower bound for constant approximating k-Clique would imply **PIH**.

Parameterized Inapproximability Hypothesis
[Lokshtanov-Ramanujan-Saurabh-Zehavi'20]:
2CSP with k variables and alphabet size n has no $(1-\epsilon)$-approximation algorithm in $f(k) \cdot n^{O(1)}$ time.

Hardness of Approximating k-Clique

- New research problem from parameterized complexity

Does k-Clique have $f(k)\cdot n^{o(k/\gamma(k))}$ time $\gamma(k)$-approximation algorithm?

- Polynomial time inapproximability does not rule out $n^{o(k/\gamma(k))}$ time algorithm.
- Assuming Gap-ETH, the answer is NO. [CCK+'17]
- It is more interesting to prove inapproximability under ETH:

From Gap-k-Clique to PIH [This Work]
An $f(k)\cdot n^{\omega(\sqrt{k})}$-time lower bound for constant approximating k-Clique would imply **PIH**.

Open problem:
Prove that ETH $\Rightarrow$ k-Clique has no $f(k)\cdot n^{o(\sqrt{k})}$ time constant approximation

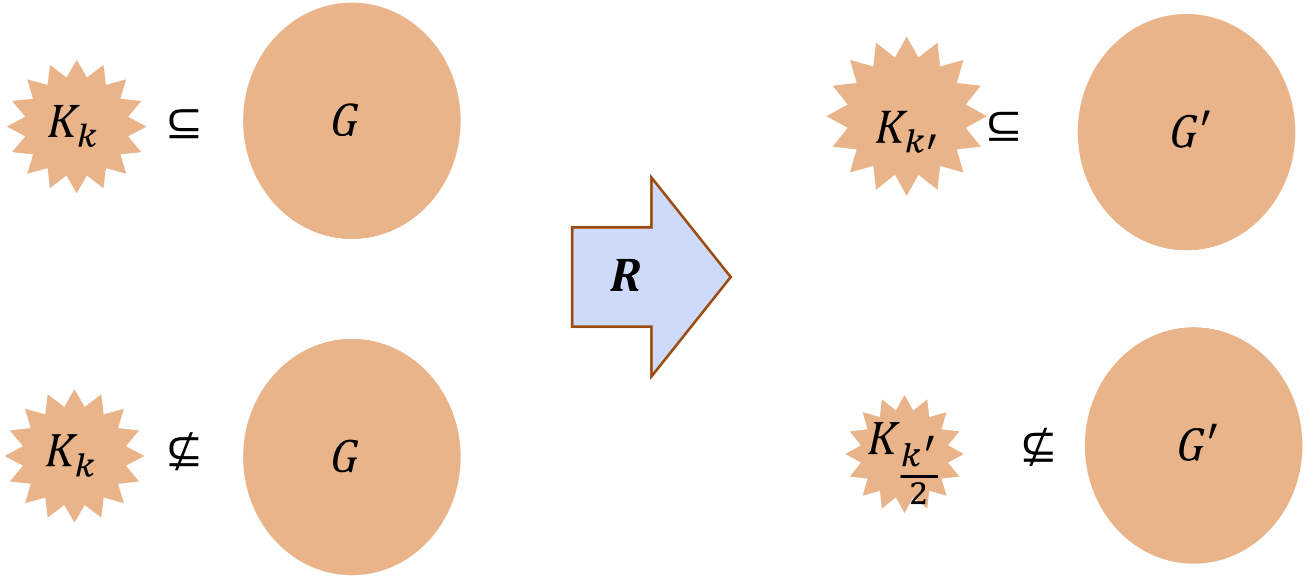
- Need a good **parameterized gap-producing reduction** for k-Clique

Parameterized Gap-producing Reduction

Parameterized Gap-producing Reduction *R*

Given a graph G and k, R outputs G' and $\boldsymbol{k'} = \boldsymbol{f}(\boldsymbol{k})$ s.t.

- If G contains a k-clique, then G' contains a k'-clique
- If G contains no k-clique, then G' contains no $(k'/2)$-clique



Parameterized Gap-producing Reduction R

Given a graph G and k, R outputs G' and $\boldsymbol{k'} = \boldsymbol{f}(\boldsymbol{k})$ s.t.

- If G contains a k-clique, then G' contains a k'-clique
- If G contains no k-clique, then G' contains no $(k'/2)$-clique

Gap

[Lin'21]

Constant Approximating k-Clique is W[1]-hard

Bingkai Lin
Nanjing University, China
lin@nju.edu.cn

- $k' = 2^{k^6}$
- ***Gap*** $= O(1)$

[Lin-Ren-Sun-Wang'22]

On Lower Bounds of Approximating Parameterized k-Clique

Bingkai Lin * Xuandi Ren† Yican Sun‡ Xiuhan Wang§

- $k' = 2^{O(k)}$
- ***Gap*** $= O(1)$

[Karthik-Khot'22]

Almost Polynomial Factor Inapproximability for Parameterized k-Clique

Karthik C. S.*
Department of Computer Science
Rutgers University
karthik.cs@rutgers.edu

Subhash Khot†
Department of Computer Science
Courant Institute of Mathematical Sciences
New York University
khot@cims.nyu.edu

- $k' = 2^{O(k^2)}$
- ***Gap*** $= (k')^{o(1)}$

[Chen-Feng-Laekhanukit-Liu'23]

Simple Combinatorial Construction of the $k^{o(1)}$-Lower Bound for Approximating the Parameterized k-Clique

Yijia Chen[1], Yi Feng[2], Bundit Laekhanukit[2], and Yanlin Liu[3]

[1]Shanghai Jiao Tong University
[2]Shanghai University of Finance and Economics
[3]Ocean University of China

- $k' = q^k$
- ***Gap*** $= q = (k')^{o(1)}$

R runs in $f(k)|G|^{O(1)}$ time

[This work]

Improved Hardness of Approximating k-Clique under ETH

Bingkai Lin * Xuandi Ren† Yican Sun‡ Xiuhan Wang§

- $k' = k^{O(\log \log k)}$
- ***Gap*** $= O(1)$

R runs in $f(k)|G|^{k^{0.54}}$ time

Overview of Previous Results

Work	Assumption	Lower Bound for Constant Approximation	Inapproximability Ratio in FPT time
[Lin'21]	W[1]≠FPT	no FPT	$O(1)$
	ETH	no $f(k) \cdot n^{o(\log^{1/6} k)}$	/
[Lin-Ren-Sun-Wang'22]	ETH	no $f(k) \cdot n^{o(\log k)}$	any $k^{o(1)}$
[Karthik-Khot'22]	W[1]≠FPT	no FPT	any $k^{o(1)}$
[Chen-Feng-Laekhanukit-Liu'23]	W[1]≠FPT	no FPT	any $k^{o(1)}$
[This work]	ETH	no $f(k) \cdot n^{k^{o(1/\log\log k)}}$	some $k^{1-o(1)}$

[This work]

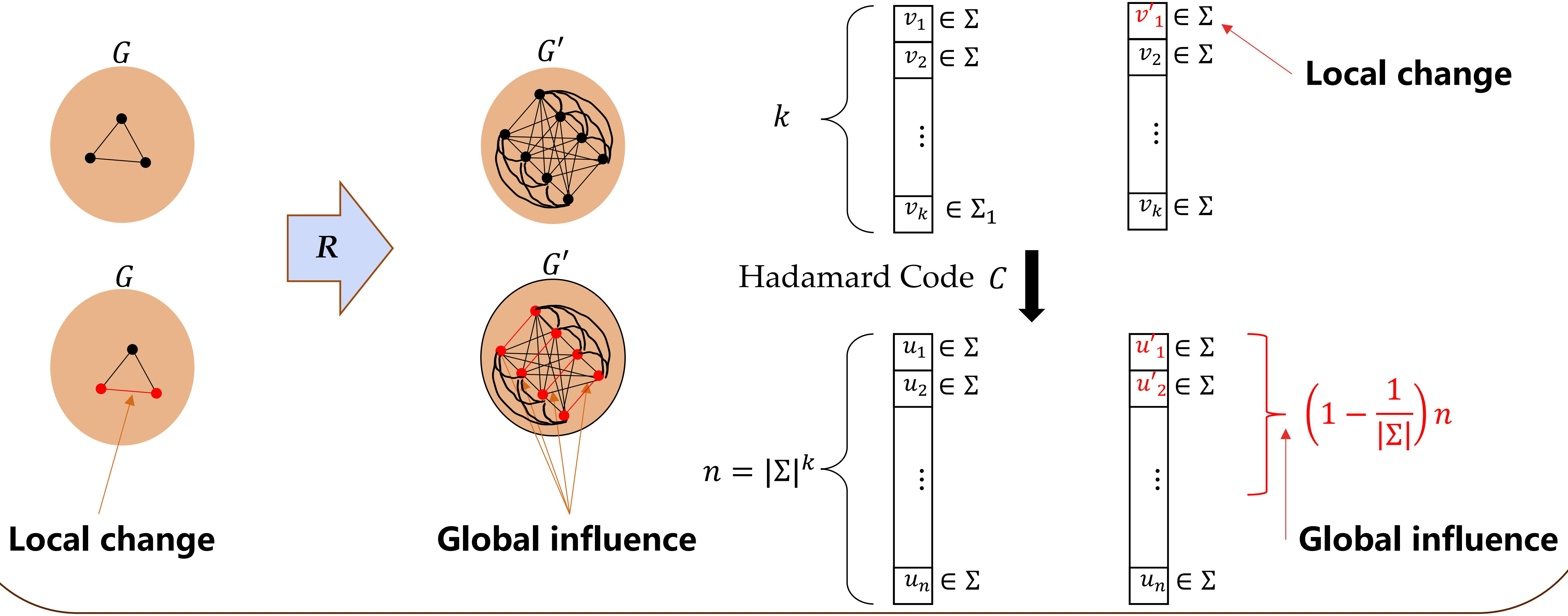
Improved Hardness of Approximating k-Clique under ETH

Bingkai Lin * Xuandi Ren† Yican Sun‡ Xiuhan Wang§

- $k' = k^{O(\log \log k)}$
- ***Gap*** $= O(1)$

R runs in $f(k)|G|^{k^{0.54}}$ time

Intuition: local to global reduction



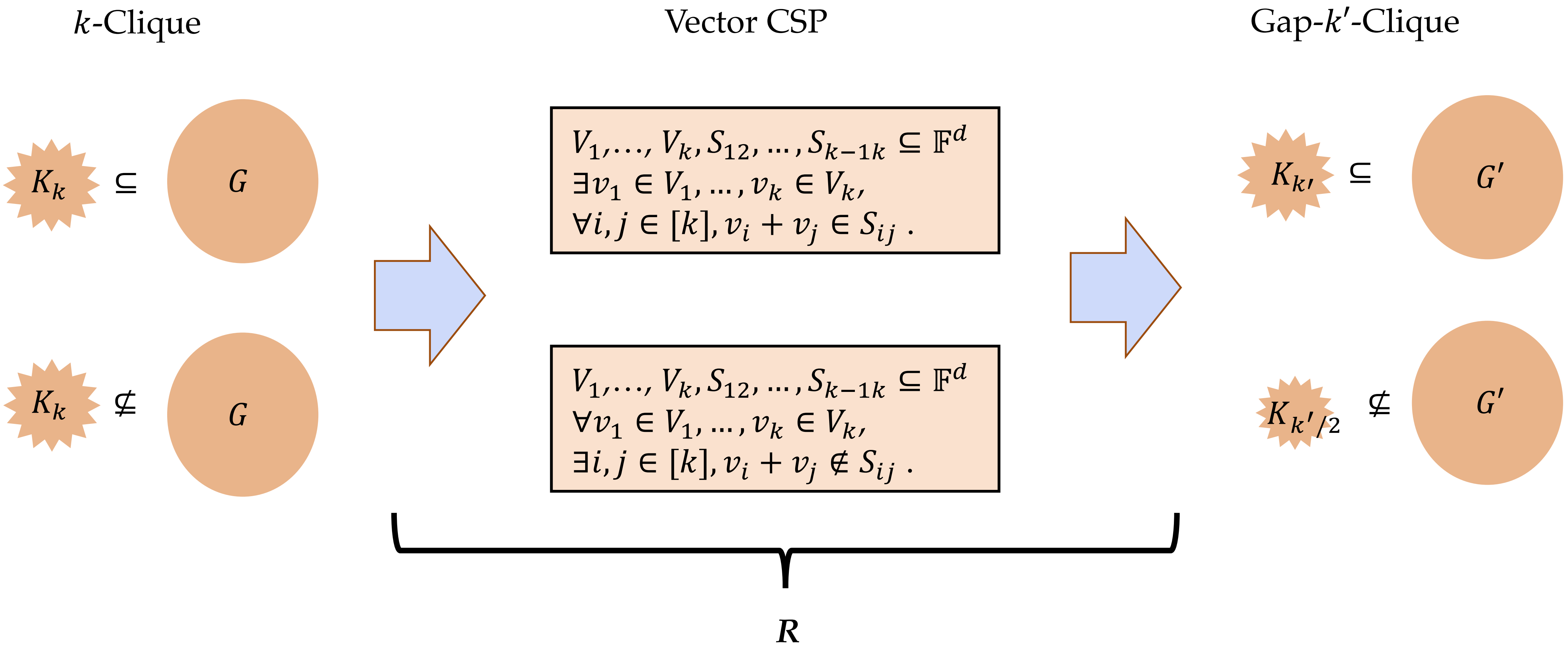
[This work]

Improved Hardness of Approximating k-Clique under ETH

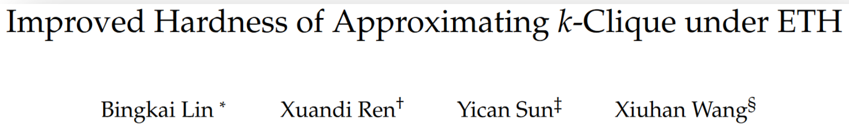
Bingkai Lin [*] Xuandi Ren [†] Yican Sun [‡] Xiuhan Wang [§]

- $k' = k^{O(\log \log k)}$
- **Gap** $= O(1)$

R runs in $f(k)|G|^{k^{0.54}}$ time



[This work]


Improved Hardness of Approximating k-Clique under ETH

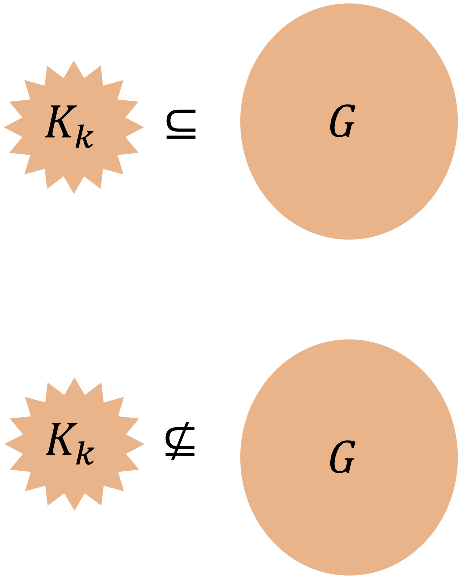
Bingkai Lin [*] Xuandi Ren[†] Yican Sun[‡] Xiuhan Wang[§]

- $k' = k^{O(\log\log k)}$
- **Gap** $= O(1)$

R runs in $f(k)|G|^{k^{0.54}}$ time

k-Clique

Vector CSP



$V_1, \ldots, V_k, S_{12}, \ldots, S_{k-1k} \subseteq \mathbb{F}^d$
$\exists v_1 \in V_1, \ldots, v_k \in V_k,$
$\forall i, j \in [k], v_i + v_j \in S_{ij}$.

- Assume $V(G) = U_1 \cup \cdots \cup U_k \subseteq \mathbb{F}^m$
- Pick random matrices $A_1, \ldots, A_k \in \mathbb{F}^{d \times m}$
- $V_i = \{A_i u : u \in U_i\}$
- $S_{ij} = \{A_i u + A_j v : u \in U_i, v \in U_j, uv \in E(G)\}$

$V_1, \ldots, V_k, S_{12}, \ldots, S_{k-1k} \subseteq \mathbb{F}^d$
$\forall v_1 \in V_1, \ldots, v_k \in V_k,$
$\exists i, j \in [k], v_i + v_j \notin S_{ij}$.

Theorem: when $\boldsymbol{d = O(\log n / \log |\mathbb{F}|)}$, w.h.p.
- $\forall$ different $u, u' \in U_i, A_i u \neq A_i u'$
- $\forall$ different $(v, u), (v', u') \in U_i \times U_j$
  $A_i v + A_j u \neq A_i v' + A_j u'$

Vector CSP with $\boldsymbol{d = O(\log n / \log |\mathbb{F}|)}$ is W[1]-hard, and has no $n^{o(k)}$ algorithms under ETH!

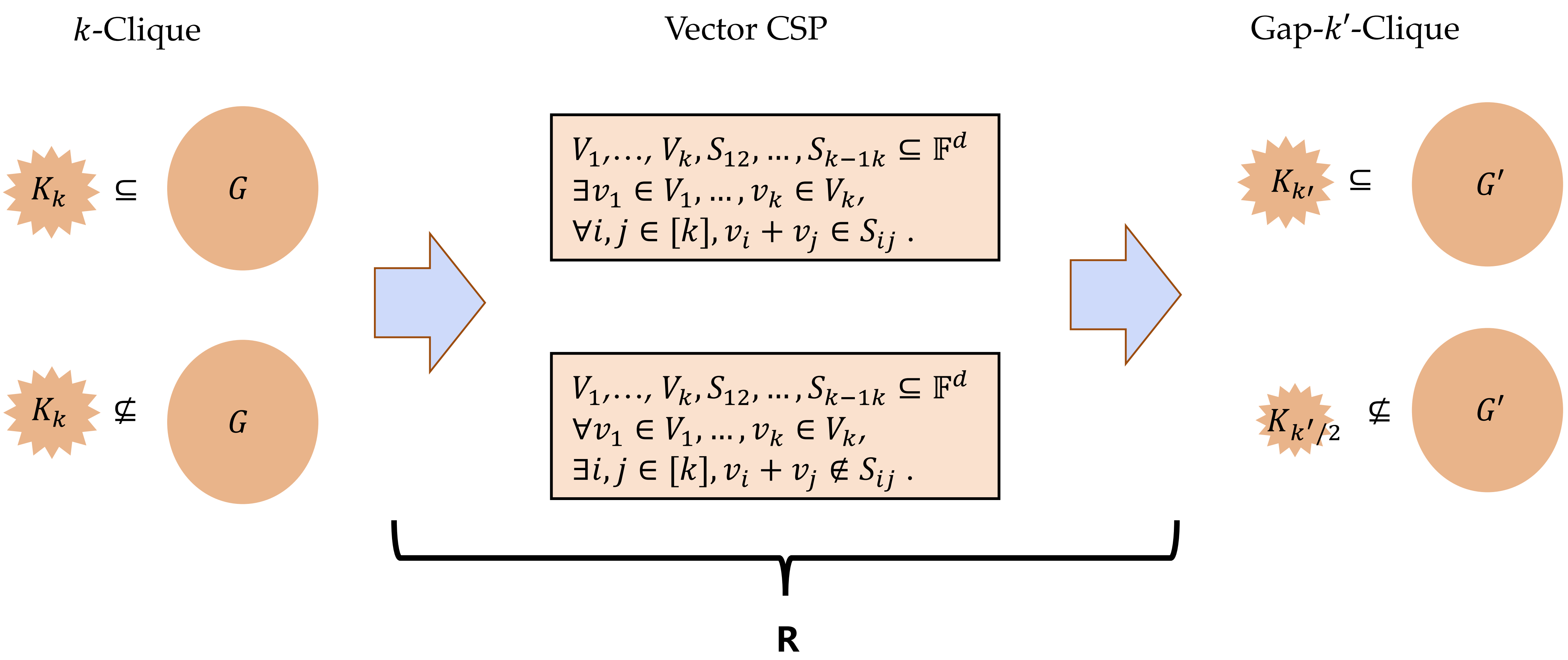
[This work]

Improved Hardness of Approximating k-Clique under ETH

Bingkai Lin * Xuandi Ren† Yican Sun‡ Xiuhan Wang§

- $k' = k^{O(\log \log k)}$
- **Gap** $= O(1)$

R runs in $f(k)|G|^{k^{0.54}}$ time



[This work]

Improved Hardness of Approximating k-Clique under ETH

Bingkai Lin [*] Xuandi Ren[†] Yican Sun[‡] Xiuhan Wang[§]

- $k' = k^{O(\log\log k)}$
- ***Gap*** $= O(1)$

R runs in $f(k)|G|^{k^{0.54}}$ time

Theorem: Given a **Parallel Locally Testable and Decodable Code**

$$C: \mathbb{F}^k \to \Sigma^{k'}$$

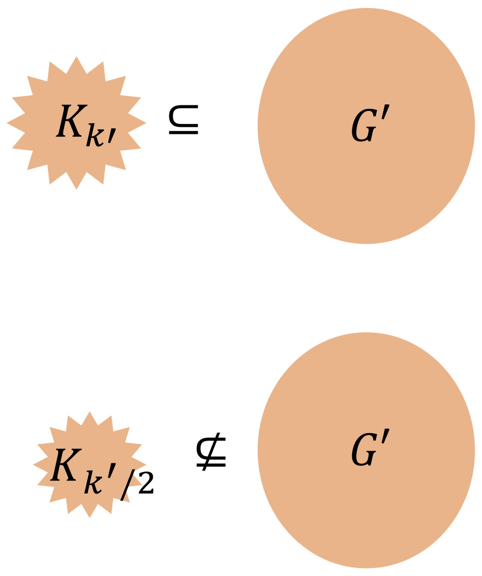
There is a reduction from **Vector CSP** to **Gap-k'-Clique**.

Vector CSP

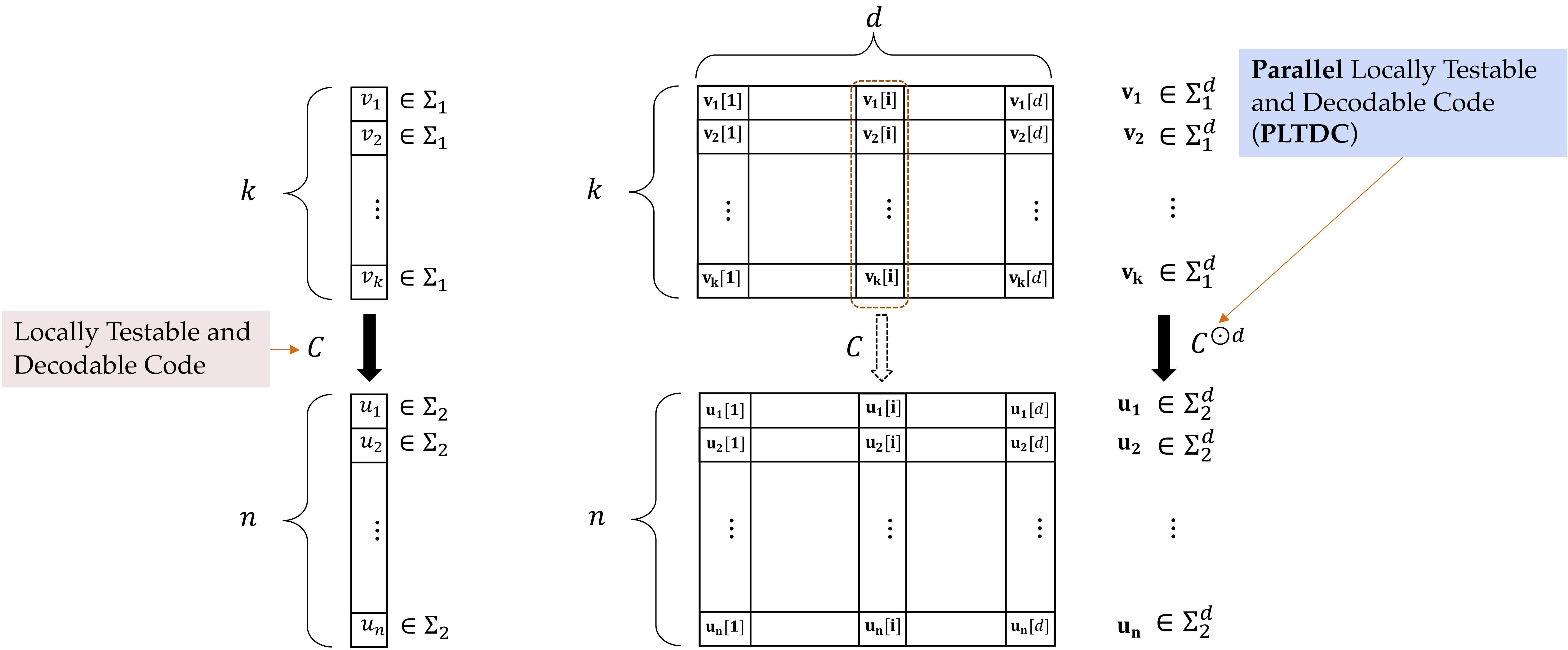
$V_1, \ldots, V_k, S_{12}, \ldots, S_{k-1k} \subseteq \mathbb{F}^d$
$\exists v_1 \in V_1, \ldots, v_k \in V_k,$
$\forall i, j \in [k], v_i + v_j \in S_{ij}$.

$V_1, \ldots, V_k, S_{12}, \ldots, S_{k-1k} \subseteq \mathbb{F}^d$
$\forall v_1 \in V_1, \ldots, v_k \in V_k,$
$\exists i, j \in [k], v_i + v_j \notin S_{ij}$.

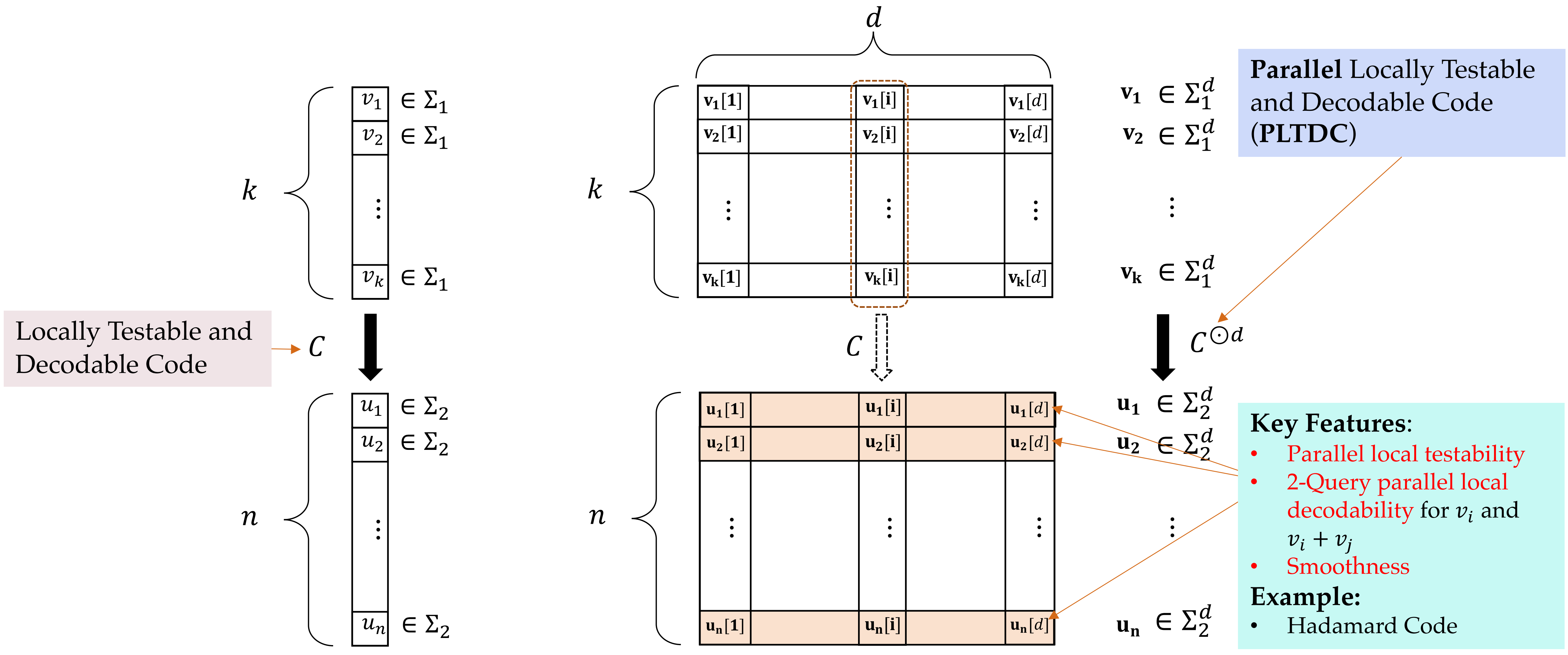
Gap-k'-Clique



Parallel Locally Testable and Decodable Code



Parallel Locally Testable and Decodable Code



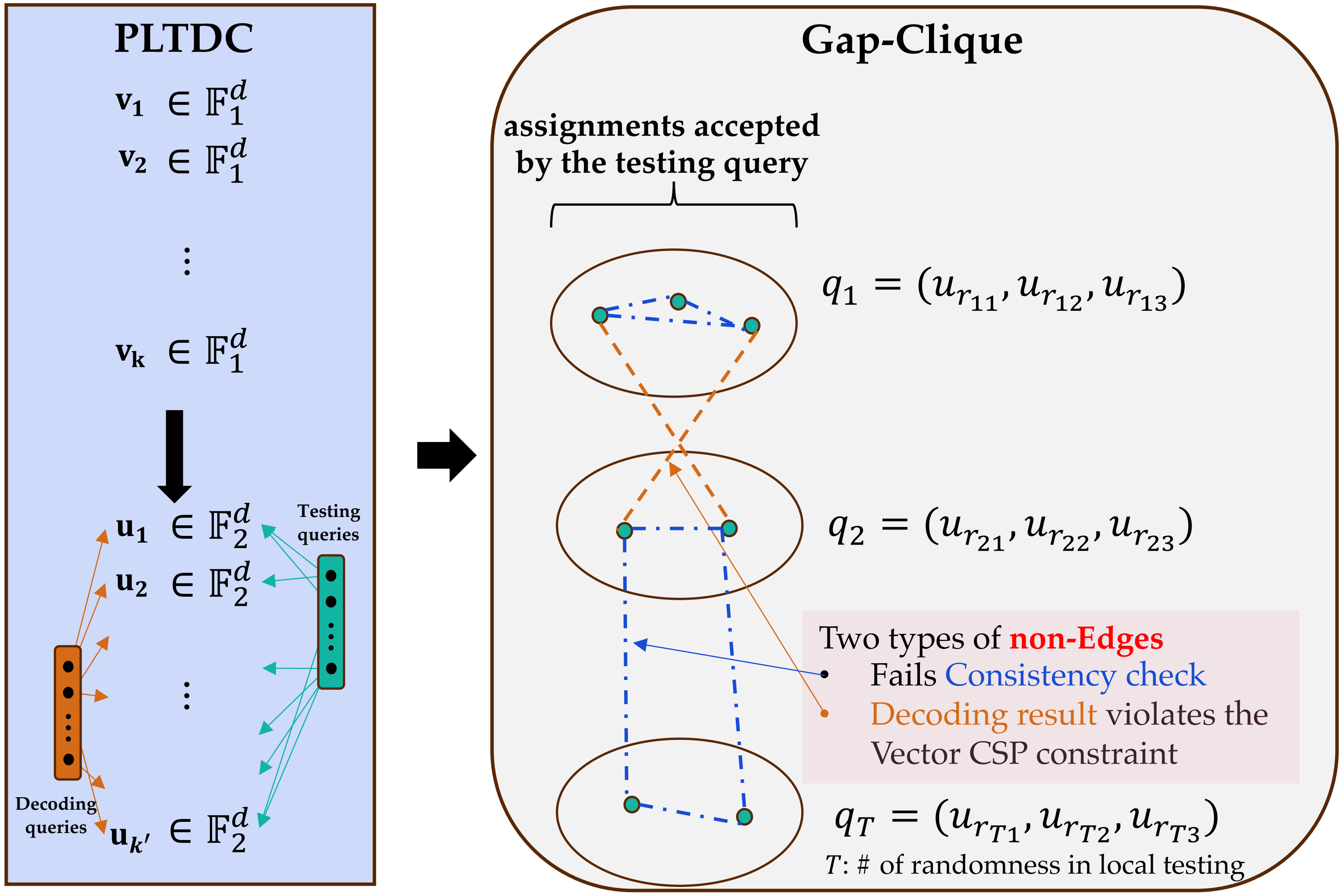
From Vec-CSP to Gap-Clique using PLTDC

Vector CSP

Input: $V_1, \ldots, V_k, S_{12}, \ldots, S_{k-1k} \subseteq \mathbb{F}_1^d$, distinguish between

(yes) $\exists v_1 \in V_1, \ldots, v_k \in V_k$,
$\forall i,j \in [k], v_i + v_j \in S_{ij}$.

(no) $\forall v_1 \in V_1, \ldots, v_k \in V_k$,
$\exists i,j \in [k], v_i + v_j \notin S_{ij}$.



PLTDC from Derivative Code

Recall PLTDC: a code mapping Σ_1^k to $\Sigma_2^{k'}$, satisfying

- Parallel local testability
- 2-Query parallel local decodability

Derivative Code [Woodruff-Yekhanin'07]: an extension of the **Reed-Muller Code**

Code	**k'**	$\mathbf{\Sigma_2}$
Hadamard Code	$(\Sigma_1)^k$	Σ_1
Derivative Code with degree 3	$(\Sigma_1)^{\sqrt[3]{k}}$	$(\Sigma_1)^{\sqrt[3]{k}+1}$
Derivative Code with degree $\Theta(\log k)$	$k^{\Theta(\log\log k)}$	$(\Sigma_1)^{k^{0.54}}$

Conclusion and Open Problem

Contributions

- A framework to prove hardness of gap k-Clique

A PLTDC is all you need!

- Improved lower bound and inapproximability ratio under ETH
  - $f(k) \cdot n^{k^{\Omega(1/\log\log k)}}$ time lower bound for constant approximation
  - $k^{1-o(1)}$ inapproximability ratio in FPT time
- Tighter connection: $f(k) \cdot n^{\omega(\sqrt{k})}$ time lower bound for constant gap k-Clique $\Rightarrow$ PIH

Open problem

- $f(k) \cdot n^{\omega(\sqrt{k})}$ time lower bound for constant gap k-Clique under ETH?

Thank You!