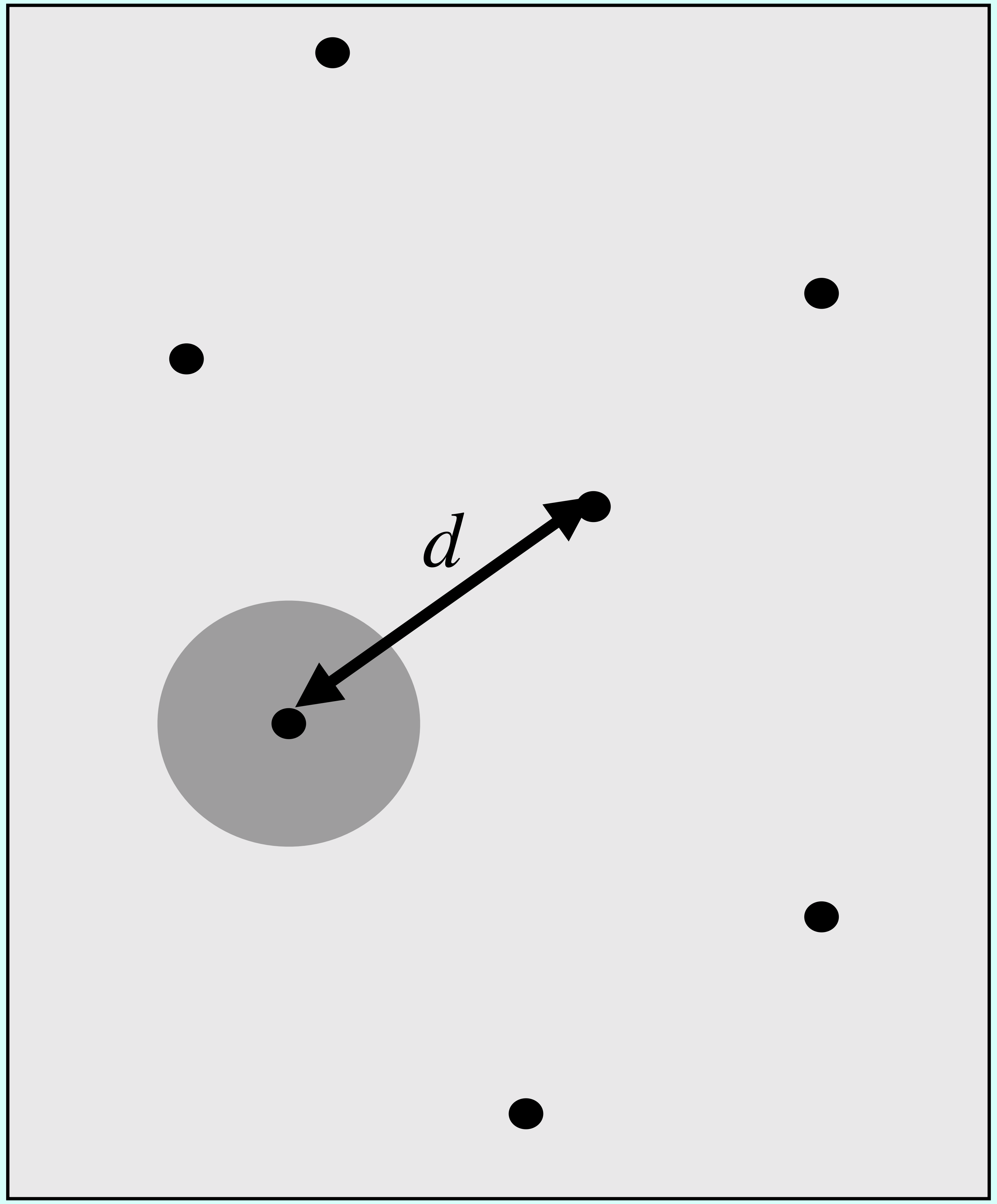
Tight Lower Bound for Approximating Parametrized Maximum Likelihood Decoding under ETH

Rishav Gupta, Bingkai Lin and Xin Zheng

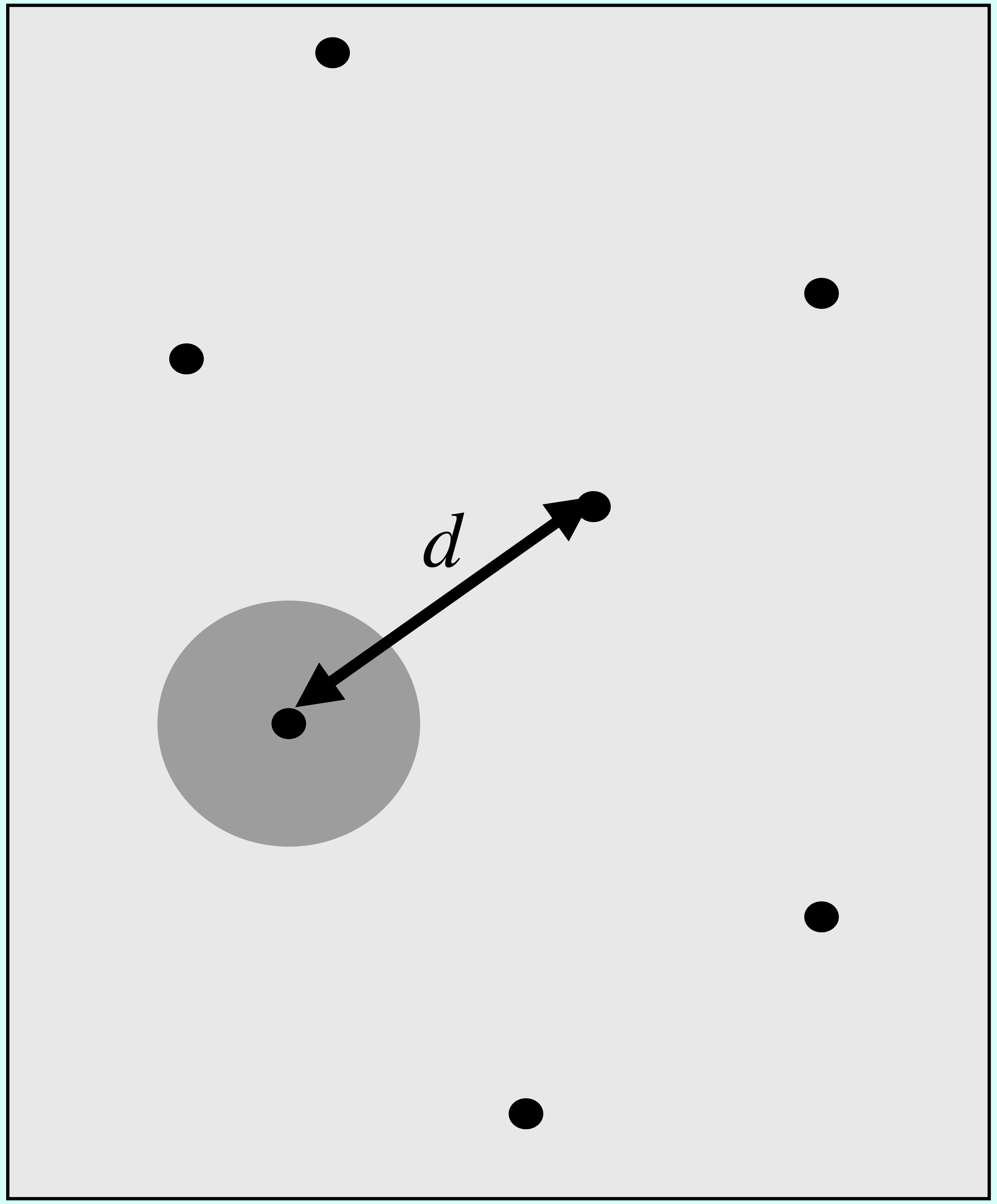
Basics of Coding Theory

What are Codes?



What are Codes?

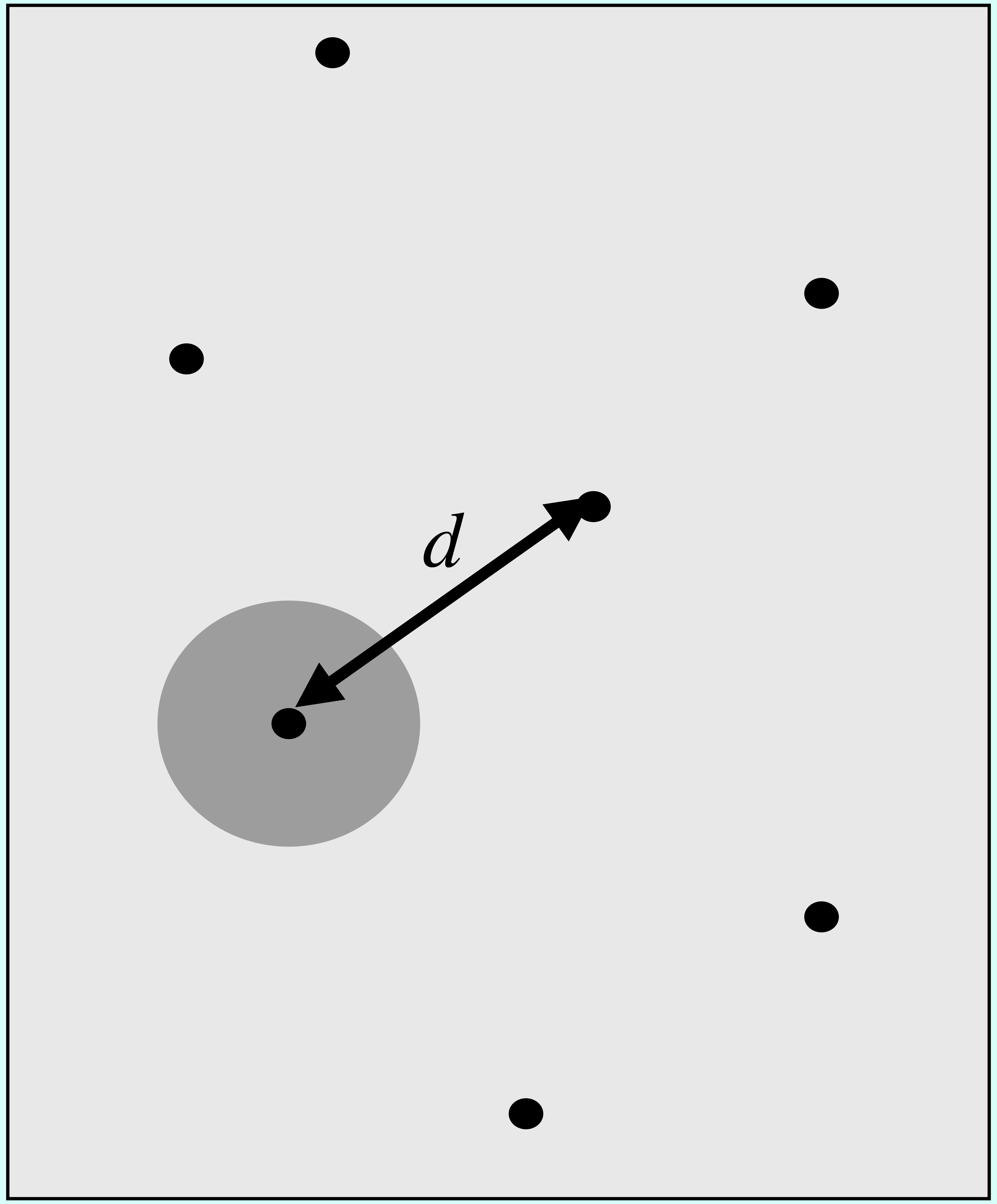
- A code is a subset of strings over some alphabet, usually represented as $C \subseteq \Sigma^m$.



What are Codes?

- A code is a subset of strings over some alphabet, usually represented as $C \subseteq \Sigma^m$.
- Distance of a code is just the minimum Hamming distance between any two codewords,

$$\mathrm{dist}(C) = \min_{x \neq y \in C} \|x - y\|_0$$



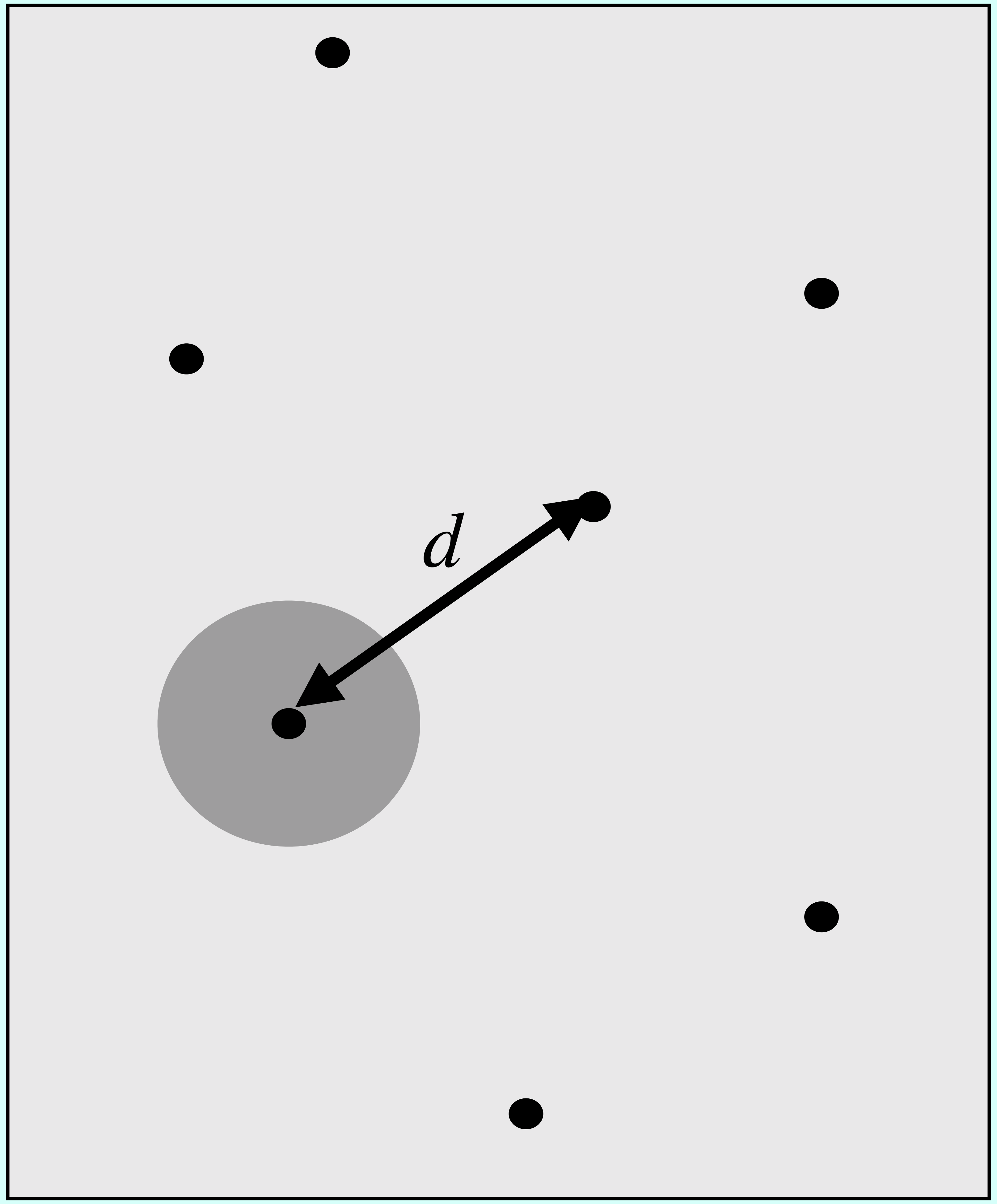
What are Codes?

- A code is a subset of strings over some alphabet, usually represented as $C \subseteq \Sigma^m$.

- Distance of a code is just the minimum Hamming distance between any two codewords,

$$\mathrm{dist}(C) = \min_{x \neq y \in C} \|x - y\|_0$$

- Linear code: A code generated by a matrix $A \in \mathbb{F}_q^{m \times n}$, $\mathrm{code}(A) = \{A \cdot x \mid x \in \mathbb{F}_q^n\}$.



What are Codes?

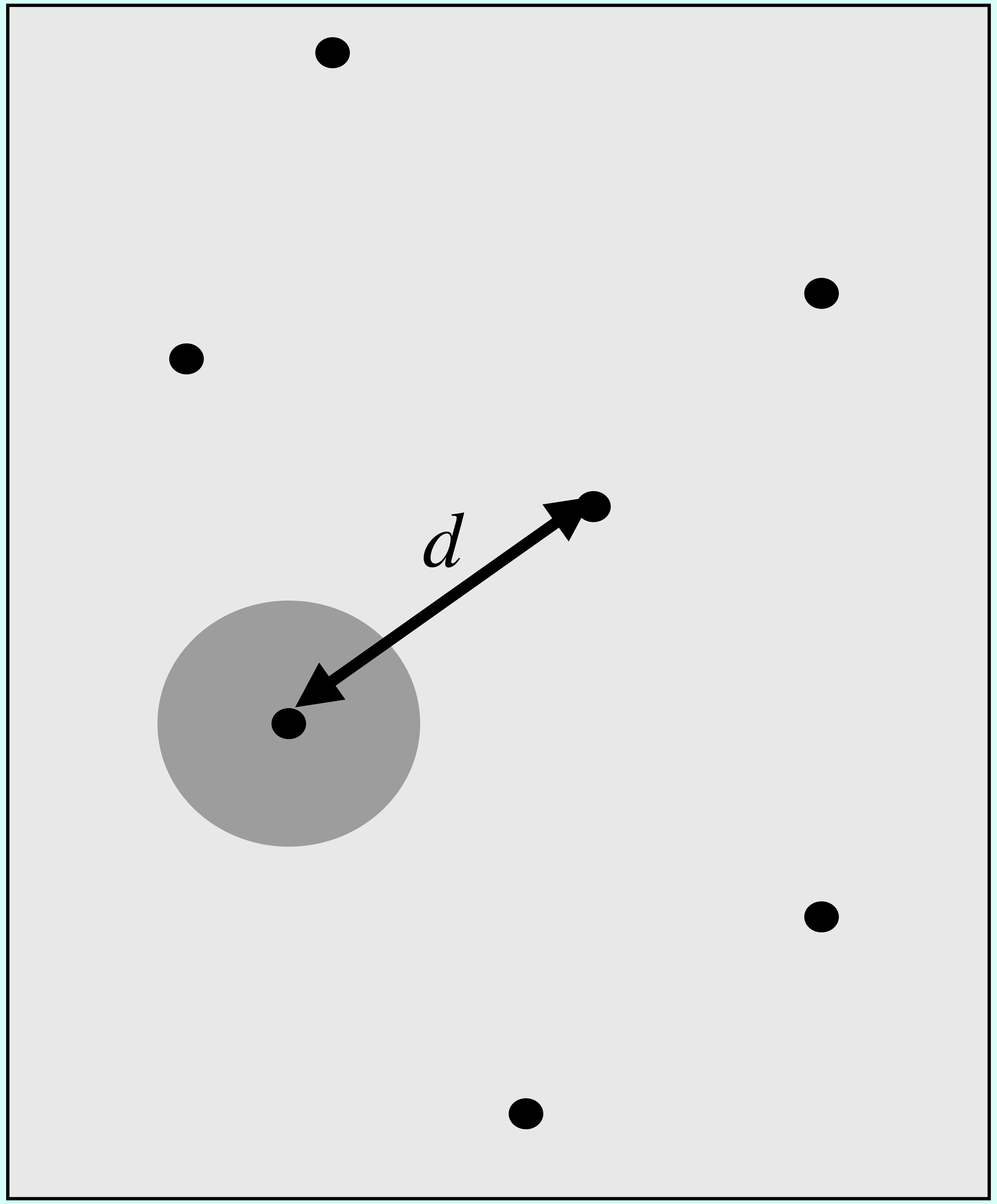
- A code is a subset of strings over some alphabet, usually represented as $C \subseteq \Sigma^m$.

- Distance of a code is just the minimum Hamming distance between any two codewords,

$$\mathrm{dist}(C) = \min_{x \neq y \in C} \|x - y\|_0$$

- Linear code: A code generated by a matrix $A \in \mathbb{F}_q^{m \times n}$, $\mathrm{code}(A) = \{A \cdot x \mid x \in \mathbb{F}_q^n\}$.

- $\mathrm{dist}(C) = \min_{x \in C \setminus \{0\}} \|x\|_0$, if C is linear.



Nearest Codeword Problem- NCP

Nearest Codeword Problem- NCP

The goal of NCP is to compute the following quantity,

$$\mathrm{dist}(y, \mathrm{code}(A)) = \min_{x \in \mathbb{F}_2^n} \|Ax - y\|_0.$$

Nearest Codeword Problem- NCP

The goal of NCP is to compute the following quantity,

$$\mathrm{dist}(y, \mathrm{code}(A)) = \min_{x \in \mathbb{F}_2^n} \|Ax - y\|_0.$$

Gap NCP: Given y, A and d distinguish between,

- **YES** Case: $\mathrm{dist}(y, \mathrm{code}(A)) \leq d.$
- **NO** Case: $\mathrm{dist}(y, \mathrm{code}(A)) > \gamma d.$

Parameterized Gap-NCP

Gap-k-NCP: Given y and A distinguish between,

- **YES** Case: $\mathrm{dist}(y, \mathrm{code}(A)) \leq k$.
- **NO** Case: $\mathrm{dist}(y, \mathrm{code}(A)) > \gamma k$.

Maximum Likelihood Decoding- MLD

Maximum Likelihood Decoding- MLD

For a parity-check matrix H and a syndrome s, the goal is to find the minimum-weight error vector e such that,

$$H \cdot e = s$$

Maximum Likelihood Decoding- MLD

For a parity-check matrix H and a syndrome s, the goal is to find the minimum-weight error vector e such that,

$$H \cdot e = s$$

Gap MLD: Given H, s and d distinguish between,

- **YES Case**: $\exists e$, such that $||e||_0 \leq d$ and $H \cdot e = s$.
- **NO Case**: $\forall e$, such that $||e||_0 \leq \gamma d \implies H \cdot e \neq s$.

Parameterized Gap-MLD

Gap-k-MLD: Given H and s distinguish between,

- **YES Case:** $\exists e$, such that $||e||_0 \leq k$ and $H \cdot e = s$.
- **NO Case:** $\forall e$, such that $||e||_0 > \gamma k \implies H \cdot e \neq s$.

NCP $\Longleftrightarrow$ MLD

$$A, b = Ax + e$$

$$\downarrow$$

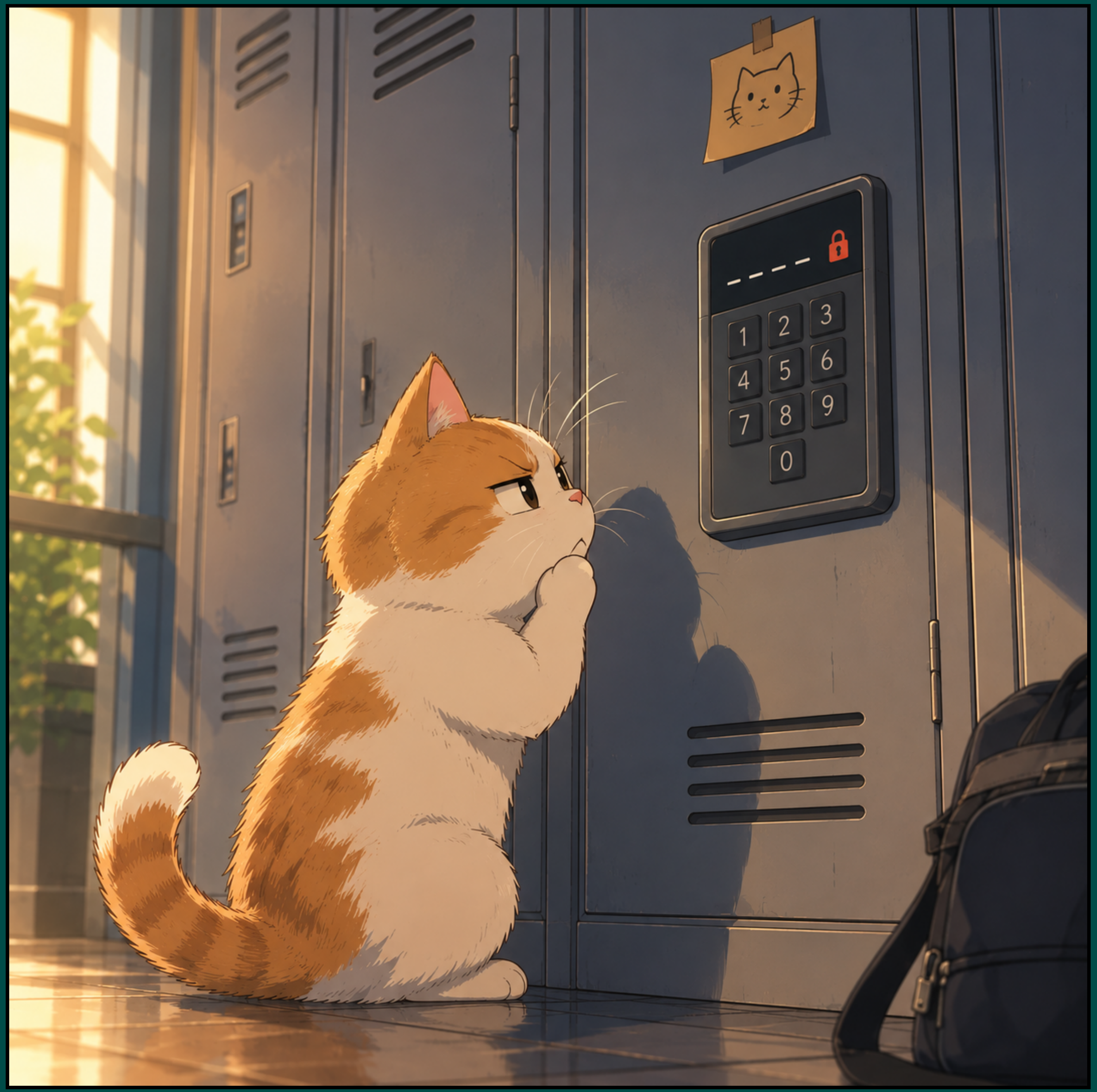
$$H, c := Hb = H \cdot Ax + He$$

Since $H \cdot A = 0$

$$\downarrow$$

$$H, He = c$$

How to Decode?



How to Decode?

How to Decode?

- Given a noisy codeword $Ax + e$, we want to decode out x.

How to Decode?

- Given a noisy codeword $Ax + e$, we want to decode out x.
- Since there are 2 choices of variables x and e, what should we brute force on?

How to Decode?

- Given a noisy codeword $Ax + e$, we want to decode out x.

- Since there are 2 choices of variables x and e, what should we brute force on?

- If we brute force on x, our time complexity is $|X| = 2^{O(n)}$.

How to Decode?

- Given a noisy codeword $Ax + e$, we want to decode out x.

- Since there are 2 choices of variables x and e, what should we brute force on?

- If we brute force on x, our time complexity is $|X| = 2^{O(n)}$.

- If we know the hamming weight of error is bounded by k then if we brute force on e, our time complexity is

$$|E| = \sum_{i=1}^{k} \binom{m}{i} \cdot q^i = m^{O(k)}.$$

How to Decode?

- Given a noisy codeword $Ax + e$, we want to decode out x.

- Since there are 2 choices of variables x and e, what should we brute force on?

- If we brute force on x, our time complexity is $|X| = 2^{O(n)}$.

- If we know the hamming weight of error is bounded by k then if we brute force on e, our time complexity is

$$|E| = \sum_{i=1}^{k} \binom{m}{i} \cdot q^i = m^{O(k)}.$$

- Hence the best brute force algorithm will switch at

$$k = O\left(\frac{n}{\log m}\right).$$

Hardness Assumptions

Exponential Time Hypothesis: ETH-[IP01]

Exponential Time Hypothesis: ETH-[IP01]

Consider an instance of 3 SAT,

$$\Phi = C_1 \wedge C_2 \wedge \cdots \wedge C_m, \qquad C_j = z_{j,1} \vee z_{j,2} \vee z_{j,3}, \qquad z_{j,k} \in \{x_i, x_i^c \mid i \in [n]\}$$

Exponential Time Hypothesis: ETH-[IP01]

Consider an instance of 3 SAT,

$$\Phi = C_1 \wedge C_2 \wedge \cdots \wedge C_m, \qquad C_j = z_{j,1} \vee z_{j,2} \vee z_{j,3}, \qquad z_{j,k} \in \{x_i, x_i^c \mid i \in [n]\}$$

- **ETH** says any algorithm deciding the satisfiblity of Φ, must take $2^{\Omega(n)}$ time.

Exponential Time Hypothesis: ETH-[IP01]

Consider an instance of 3 SAT,

$$\Phi = C_1 \wedge C_2 \wedge \cdots \wedge C_m, \qquad C_j = z_{j,1} \vee z_{j,2} \vee z_{j,3}, \qquad z_{j,k} \in \{x_i, x_i^c \mid i \in [n]\}$$

- **ETH** says any algorithm deciding the satisfiblity of Φ, must take $2^{\Omega(n)}$ time.
- **ETH** (randomized) is same as **ETH** but also infers all randomized algorithms.

Hardness of Gap-NCP [BHIRW24]

Hardness of Gap-NCP [BHIRW24]

A reduction from 3SAT with n_0 variables to Gap-NCP instance $A_{m \times n}, b$ over $\mathbb{F}_q$ with $n = \mathcal{O}(n_0)$, $m = \mathcal{O}(n_0)$, and some constants $\alpha, \gamma > 1$.

Hardness of Gap-NCP [BHIRW24]

A reduction from 3SAT with n_0 variables to Gap-NCP instance $A_{m \times n}, b$ over $\mathbb{F}_q$ with $n = \mathcal{O}(n_0)$, $m = \mathcal{O}(n_0)$, and some constants $\alpha, \gamma > 1$.

- **YES Case:** $\exists x = (x_1, \ldots, x_n) \in \mathbb{F}_q^n$ such that, $\|Ax - b\|_0 \leq \alpha \cdot m$.

Hardness of Gap-NCP [BHIRW24]

A reduction from 3SAT with n_0 variables to Gap-NCP instance $A_{m \times n}, b$ over $\mathbb{F}_q$ with $n = \mathcal{O}(n_0)$, $m = \mathcal{O}(n_0)$, and some constants $\alpha, \gamma > 1$.

- **YES Case:** $\exists x = (x_1, \ldots, x_n) \in \mathbb{F}_q^n$ such that, $\|Ax - b\|_0 \leq \alpha \cdot m$.
- **NO Case:** $\forall x = (x_1, \ldots, x_n) \in \mathbb{F}_q^n$, such that, $\|Ax - b\|_0 > \gamma \cdot \alpha \cdot m$.

Hardness of Gap-MLD [BHIRW24]

Hardness of Gap-MLD [BHIRW24]

A reduction from 3SAT with n_0 variables to Gap-MLD instance $H_{m-n \times m}, c$ over $\mathbb{F}_q$ with $n = \mathcal{O}(n_0)$ variables and $m = \mathcal{O}(n_0)$ and some constants $\alpha, \gamma > 1$,

Hardness of Gap-MLD [BHIRW24]

A reduction from 3SAT with n_0 variables to Gap-MLD instance $H_{m-n \times m}, c$ over $\mathbb{F}_q$ with $n = \mathcal{O}(n_0)$ variables and $m = \mathcal{O}(n_0)$ and some constants $\alpha, \gamma > 1$,

- **YES Case**: $\exists \ \|e\|_0 \leq \alpha \cdot m$ and $He = c$.

Hardness of Gap-MLD [BHIRW24]

A reduction from 3SAT with n_0 variables to Gap-MLD instance $H_{m-n \times m}, c$ over $\mathbb{F}_q$ with $n = \mathcal{O}(n_0)$ variables and $m = \mathcal{O}(n_0)$ and some constants $\alpha, \gamma > 1$,

- **YES Case:** $\exists \ \|e\|_0 \leq \alpha \cdot m$ and $He = c$.
- **NO Case:** $\|e\|_0 \leq \gamma \cdot \alpha \cdot m \implies He \neq c$.

Our Results

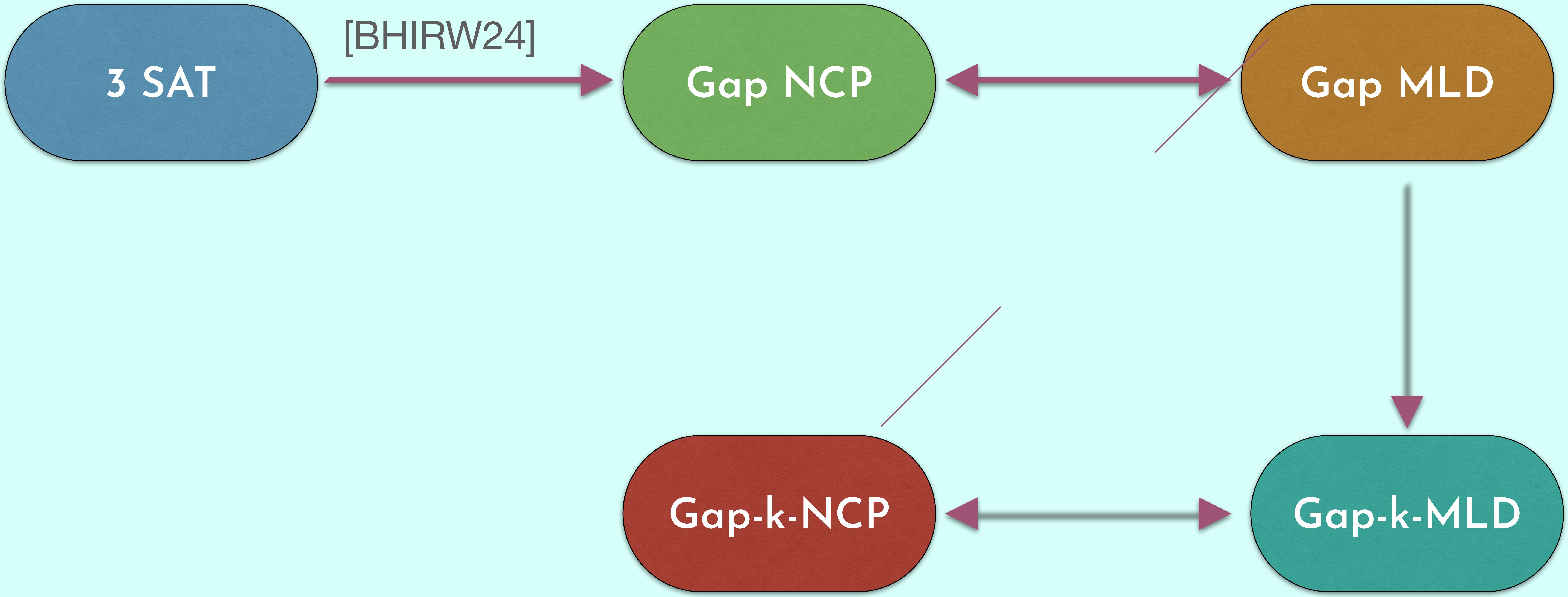

Study
Hard
Be
Kind
MATHEMATICS
PHYSICS
CHEMISTRY
BIOLOGY
HISTORY

Hardness of Gap-k-NCP/MLD

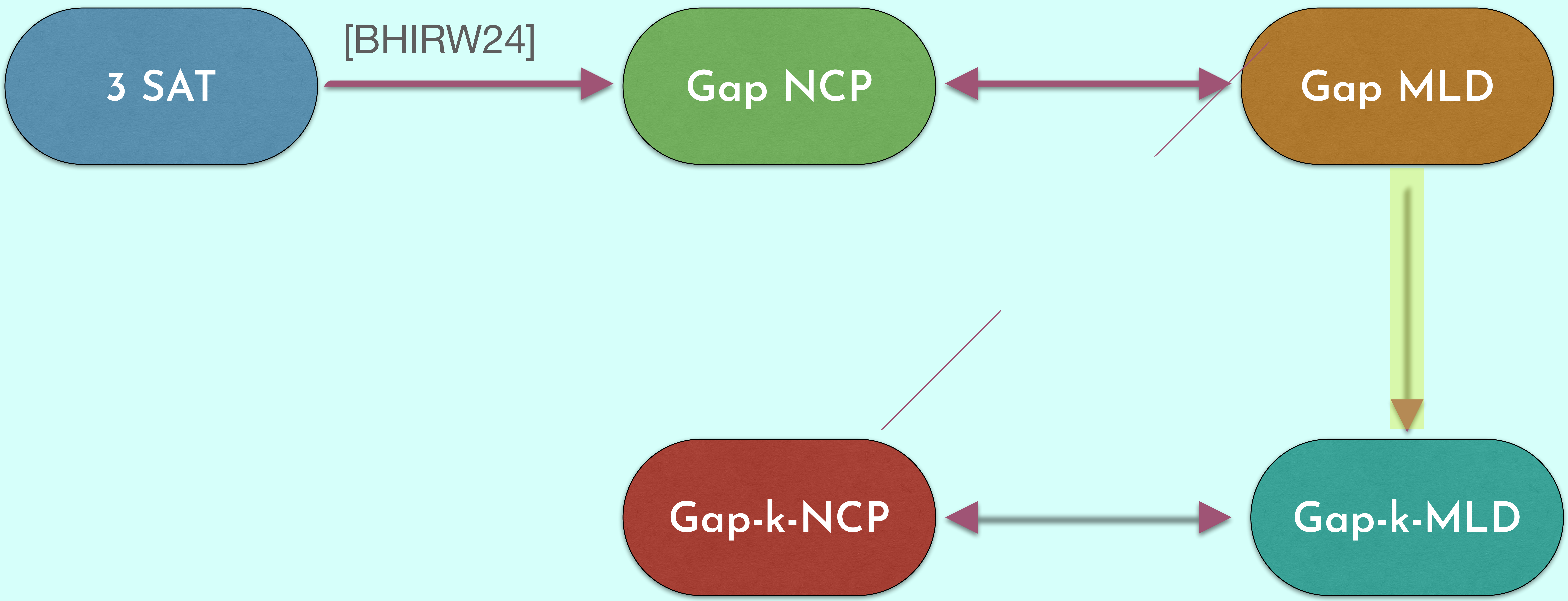
Result	Assumption	Lower bound	Approx. factor
[Man20]	randomized Gap-ETH	No $m^{o(k)}$ algorithm	all constant factors
[BKM25]	ETH	No $m^{o(k/\log^c k)}$ algorithm	all constant factors
Our result	ETH	No $m^{o(k)}$ algorithm	some constant factor
Our result (informal)	randomized ETH	Brute-force optimal; no $m^{o(k)}$ even when $k \leq m^{\varepsilon}$	some constant factor

Reduction Sketch

Reduction Sketch



Reduction Sketch



Reduction Sketch

Reduction Sketch

- We will reduce the **Gap-MLD** instance from [BHIRW24] to **Gap-k-MLD** instance.

Reduction Sketch

- We will reduce the **Gap-MLD** instance from [BHIRW24] to **Gap-k-MLD** instance.
- So the **Gap-MLD** instance from [BHIRW24] looks like,

Reduction Sketch

- We will reduce the **Gap-MLD** instance from [BHIRW24] to **Gap-k-MLD** instance.
- So the **Gap-MLD** instance from [BHIRW24] looks like,

Reduction Sketch

- We will reduce the **Gap-MLD** instance from [BHIRW24] to **Gap-k-MLD** instance.
- So the **Gap-MLD** instance from [BHIRW24] looks like,

$$\underbrace{\begin{bmatrix} | & | & & | \\ h_1 & h_2 & \cdots & h_m \\ | & | & & | \end{bmatrix}}_{H \in \mathbb{F}_q^{m-n \times m}} \cdot e = c,$$

- **YES Case:** c is in span of some αm $h_i's$.
- **NO Case:** c is not in the span of any $(\alpha + \varepsilon)m$ $h_i's$.

Reduction Sketch

Reduction Sketch

$$\underbrace{\begin{bmatrix} | & | & & | \\ h_1 & h_2 & \cdots & h_m \\ | & | & & | \end{bmatrix}}_{H \in \mathbb{F}_q^{m-n \times m}} \cdot e = c,$$

- **YES Case:** c is in span of some αm $h_i's$.
- **NO Case:** c is not in the span of any $\gamma \alpha m$ $h_i's$.

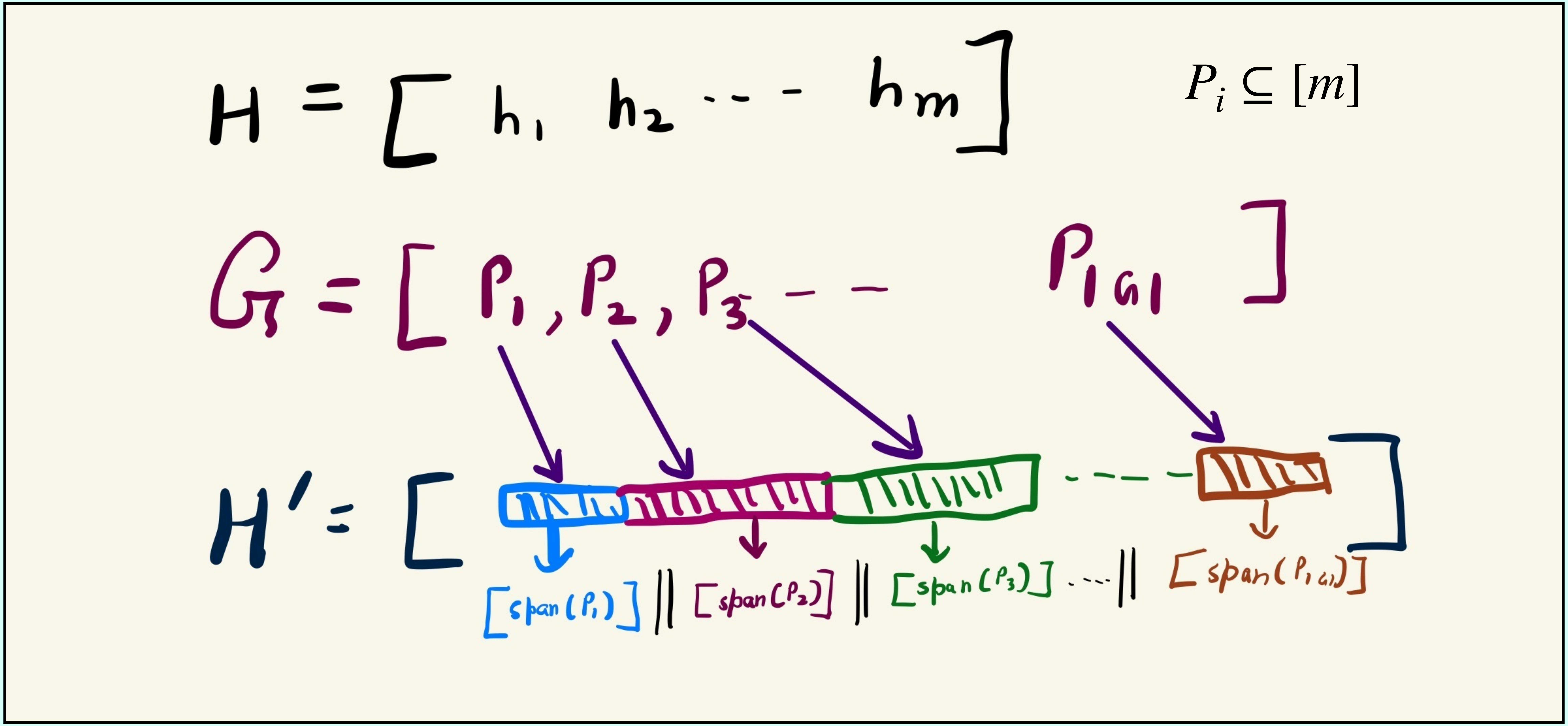
Reduction Sketch

$$\underbrace{\begin{bmatrix} | & | & & | \\ h_1 & h_2 & \cdots & h_m \\ | & | & & | \end{bmatrix}}_{H \in \mathbb{F}_q^{m-n \times m}} \cdot e = c,$$

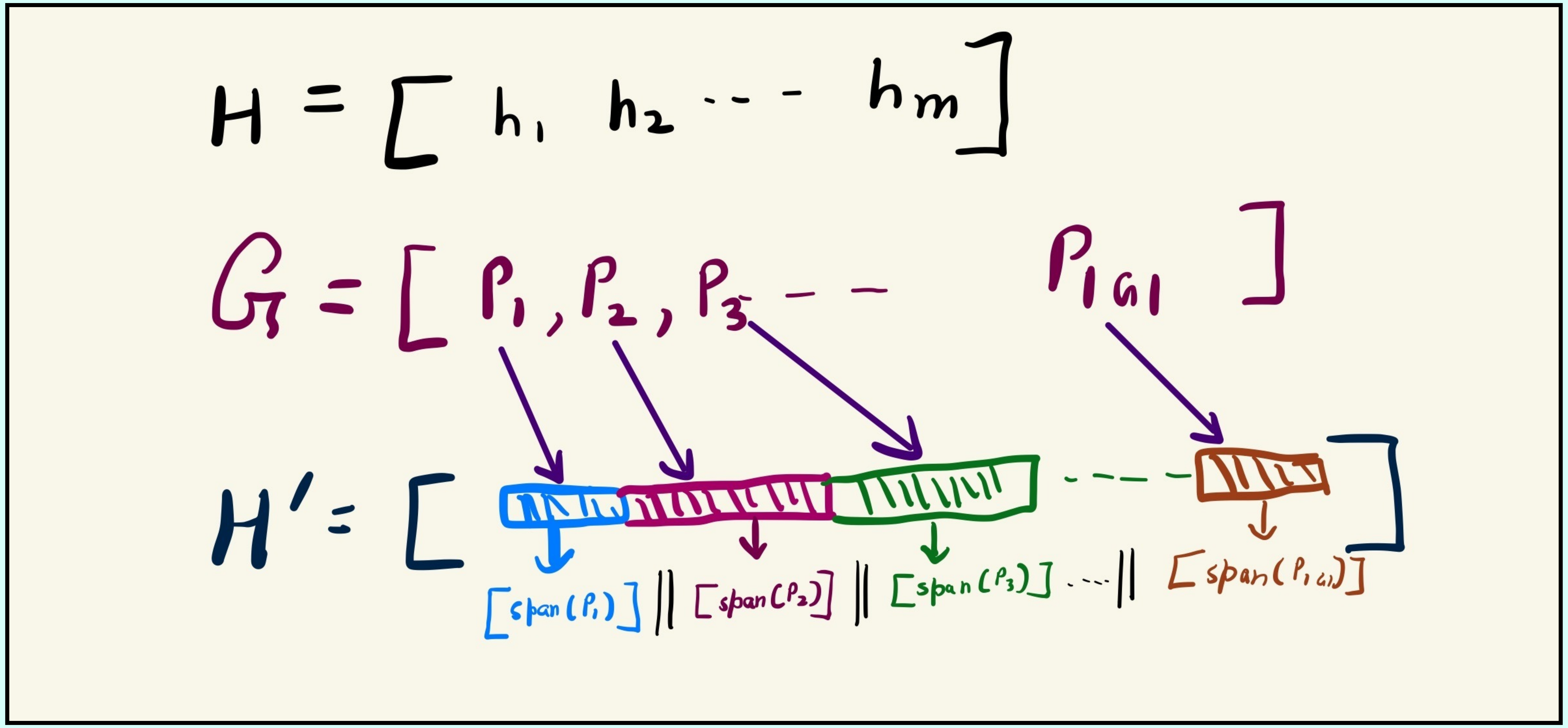
- **YES Case:** c is in span of some αm h_i's.
- **NO Case:** c is not in the span of any $\gamma \alpha m$ h_i's.

- To get parametrized hardness we would want to take the above as input, and a parameter k and generate H' and c' such that,
  - **YES Case:** c' is in the span of some k columns of H'.
  - **NO Case:** c' is not in the span of any γk columns of H'.

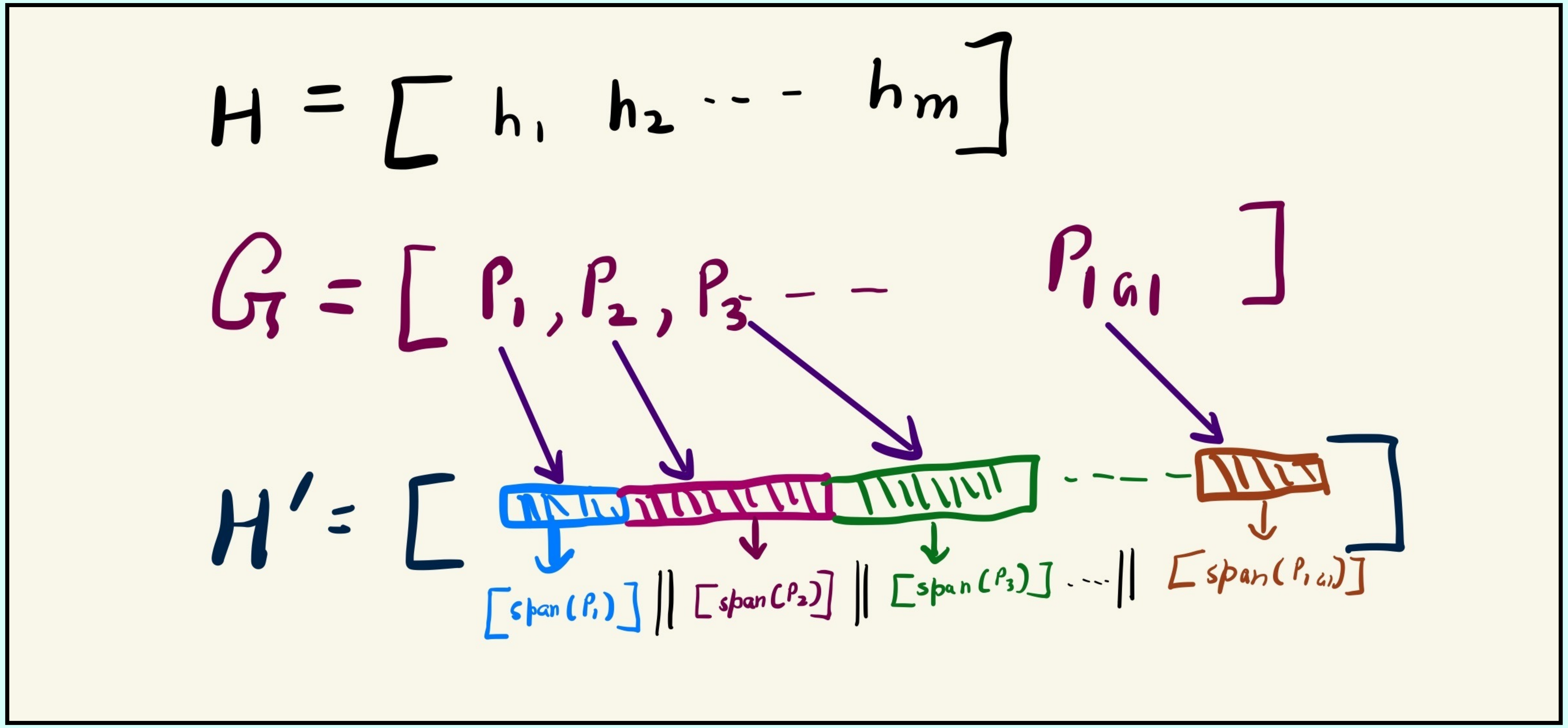
Grouping Reduction



Naive Grouping Reduction

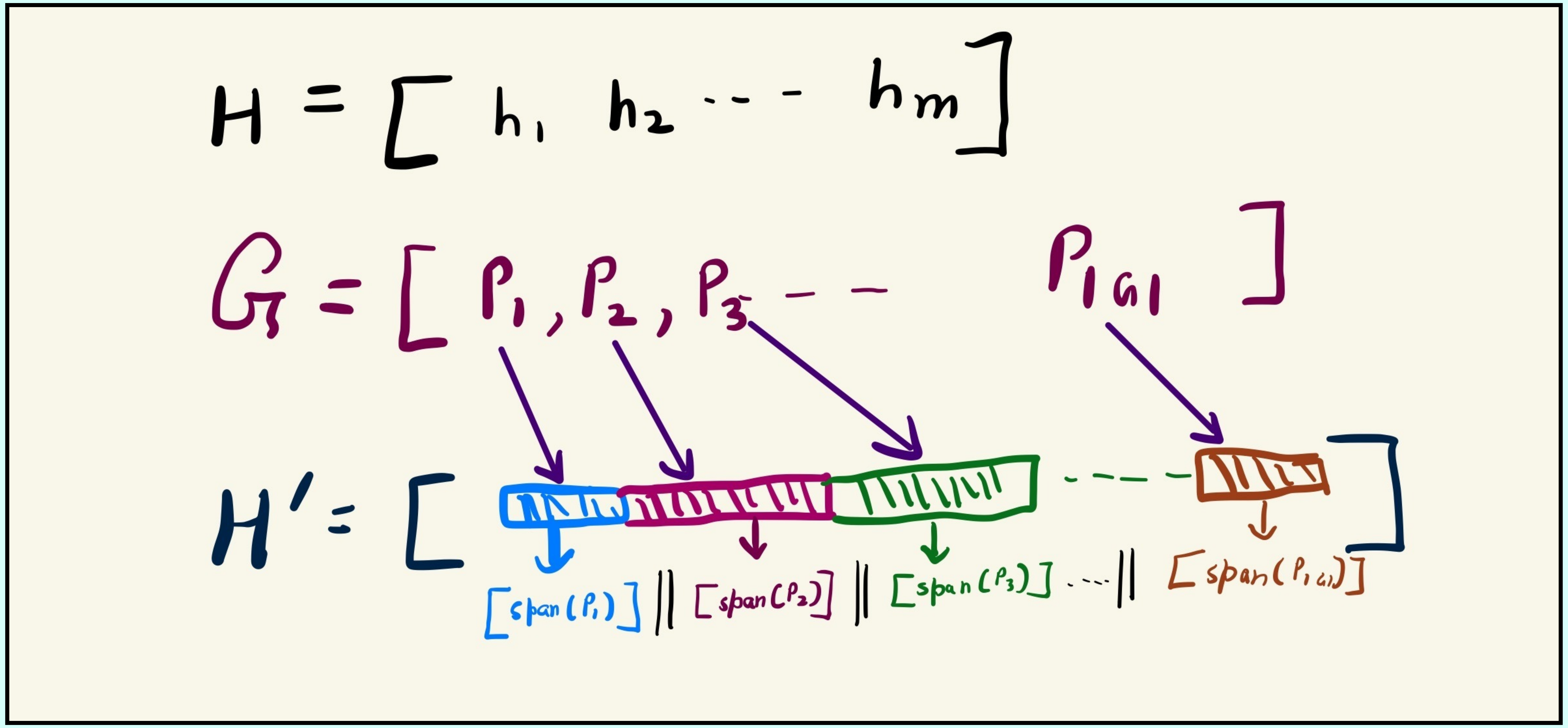


Naive Grouping Reduction

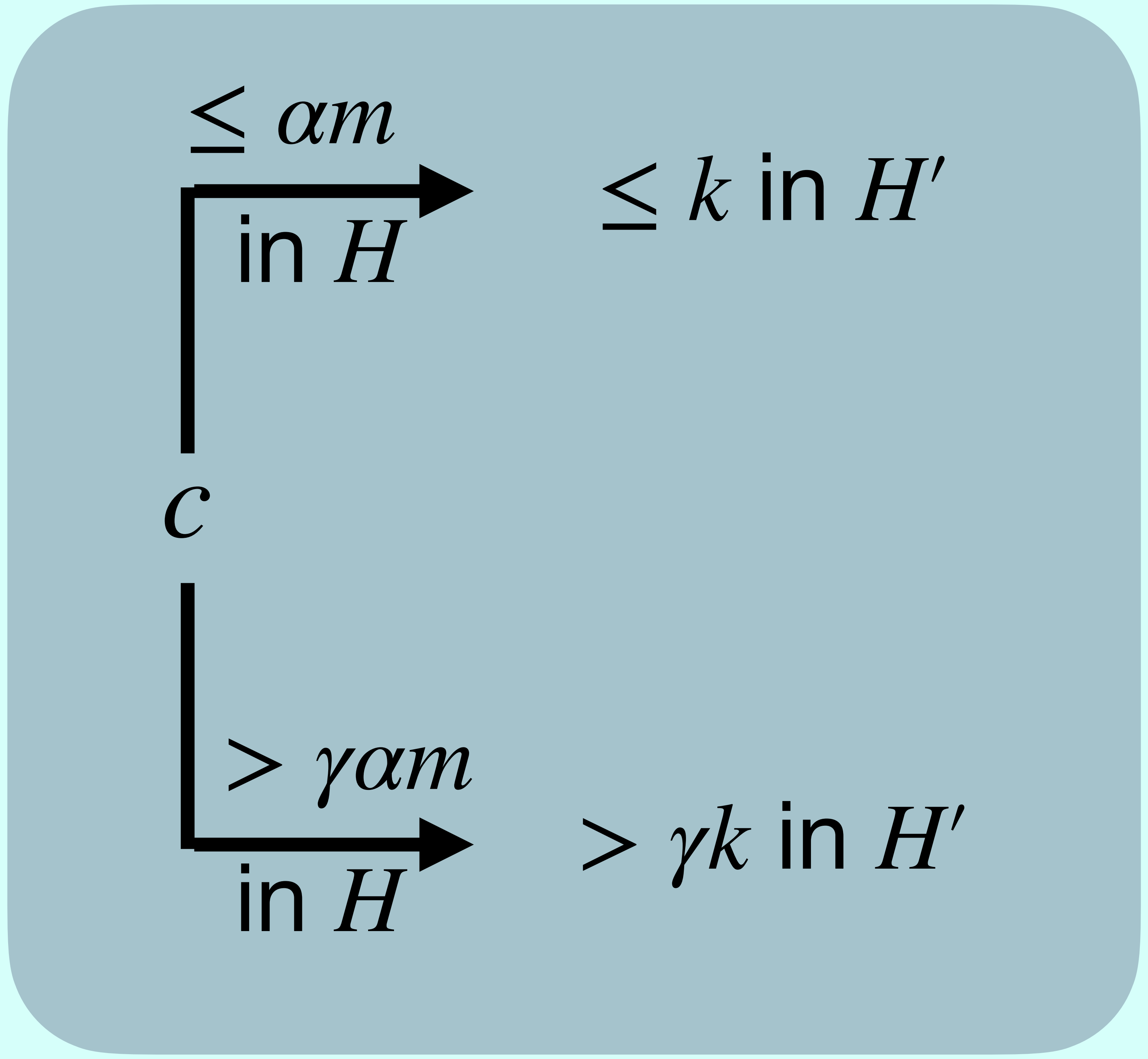


- Take G to be all collection of all subsets of size $\alpha m/k$ of $[m]$.

Naive **Grouping** Reduction

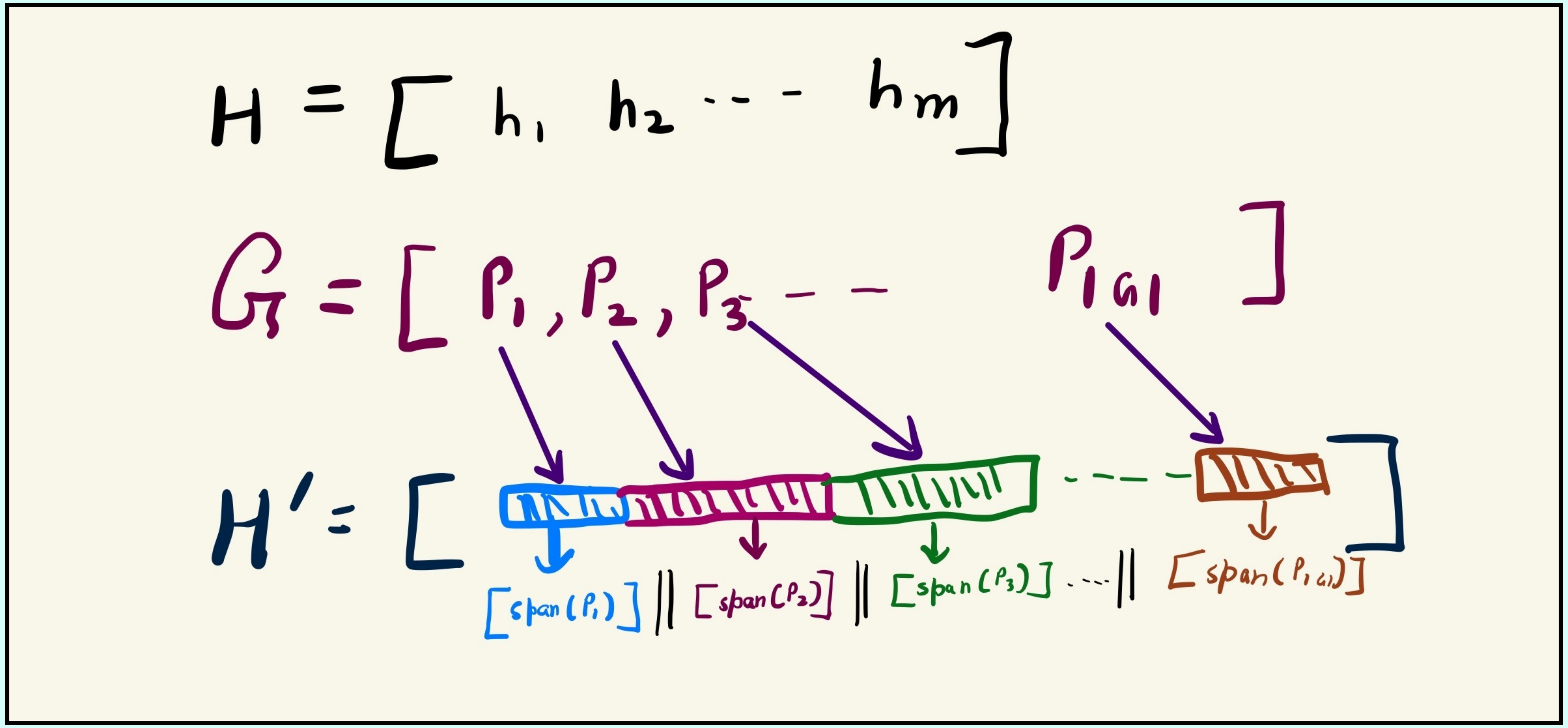


- Take G to be all collection of all subsets of size $\alpha m/k$ of $[m]$.

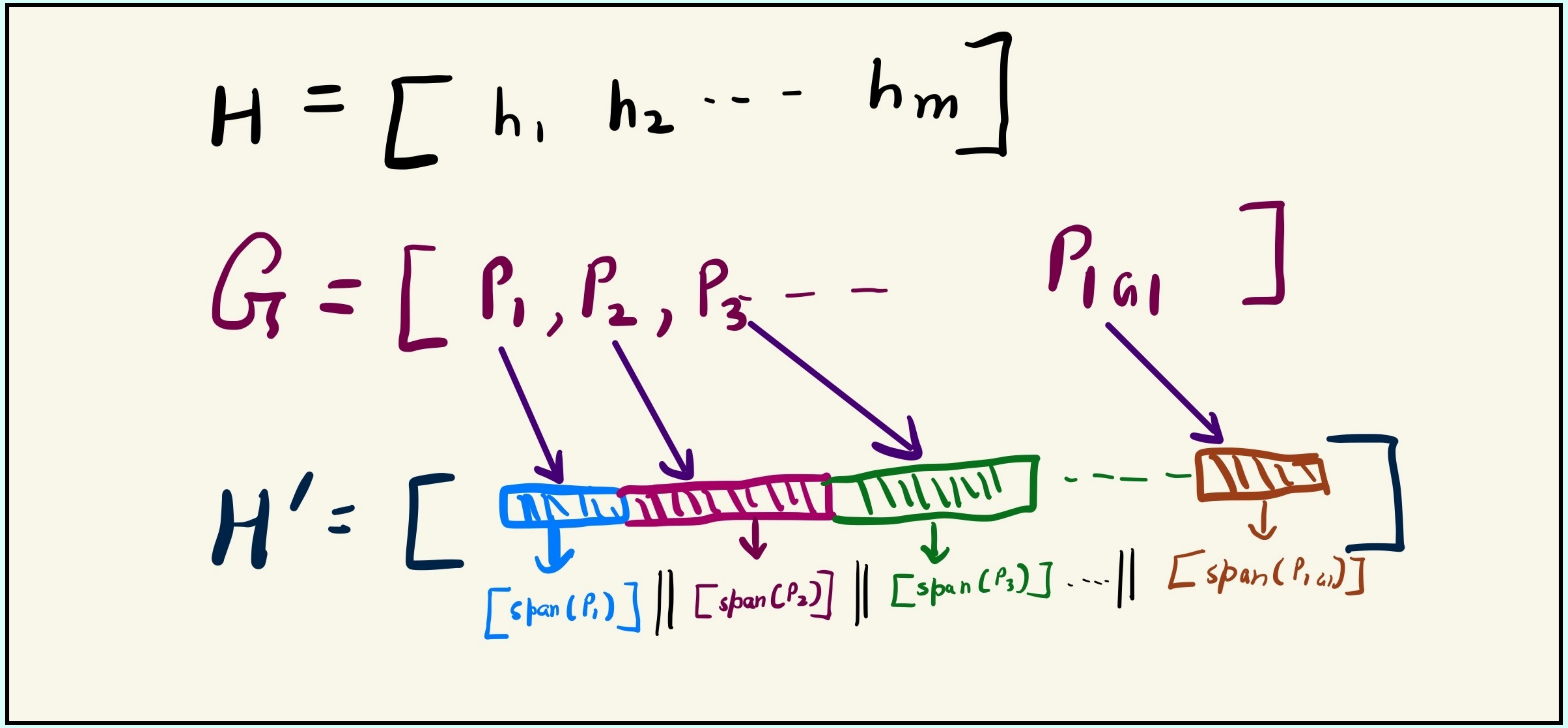


$$
\begin{aligned}
c = {} & \underbrace{x_1 h_1 + \cdots + x_{\alpha m/k} h_{\alpha m/k}}_{\in \operatorname{span}(P_1)} + \\
& \underbrace{x_{\alpha m/k+1} h_{\alpha m/k+1} + \cdots + x_{2\alpha m/k} h_{2\alpha m/k}}_{\in \operatorname{span}(P_2)} + \cdots + \\
& \underbrace{x_{(k-1)\alpha m/k+1} h_{(k-1)\alpha m/k+1} + \cdots + x_{\alpha m} h_{\alpha m}}_{\in \operatorname{span}(P_k)} .
\end{aligned}
$$

Naive Grouping Reduction

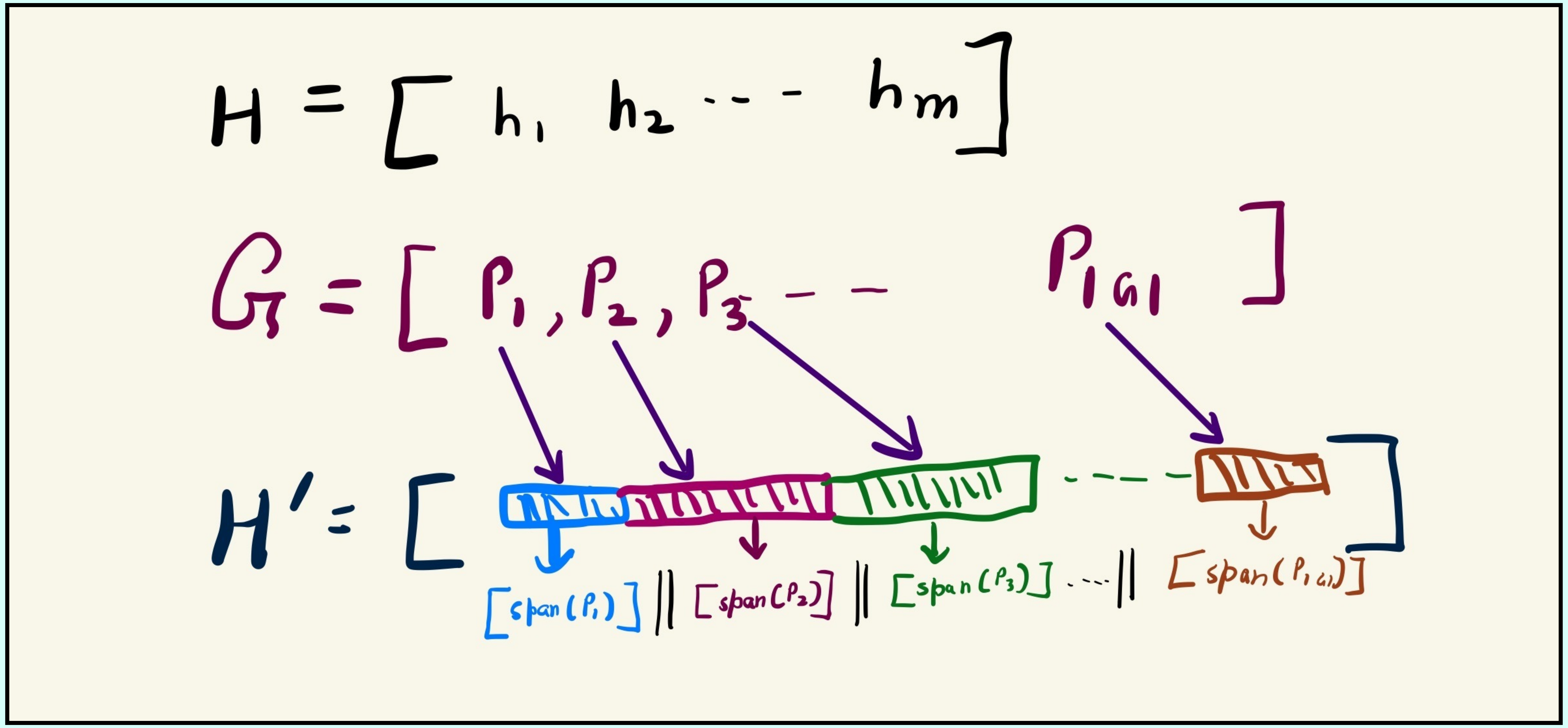


Naive Grouping Reduction

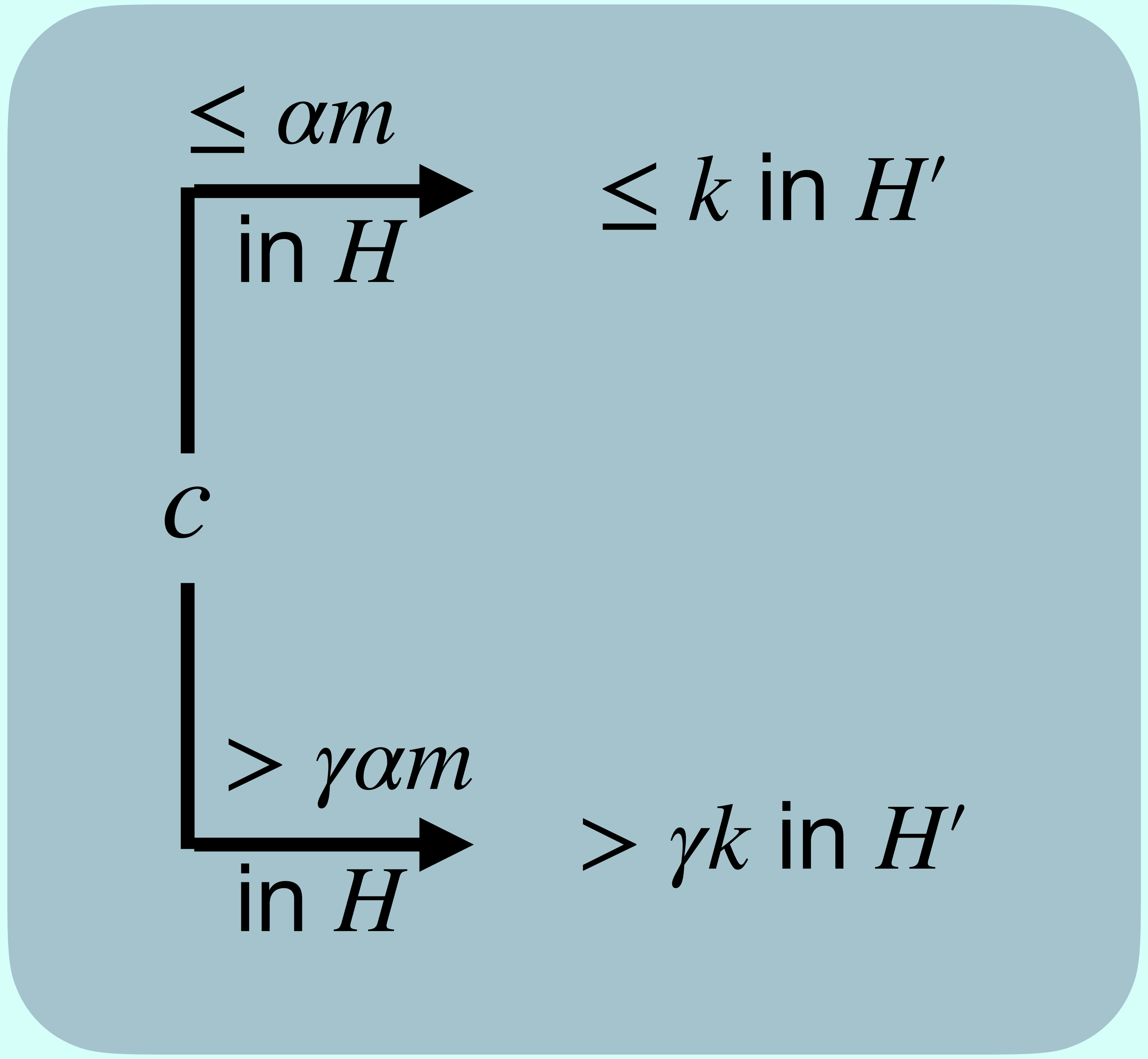


- Take G to be all collection of all subsets of size $\alpha m/k$ of $[m]$.

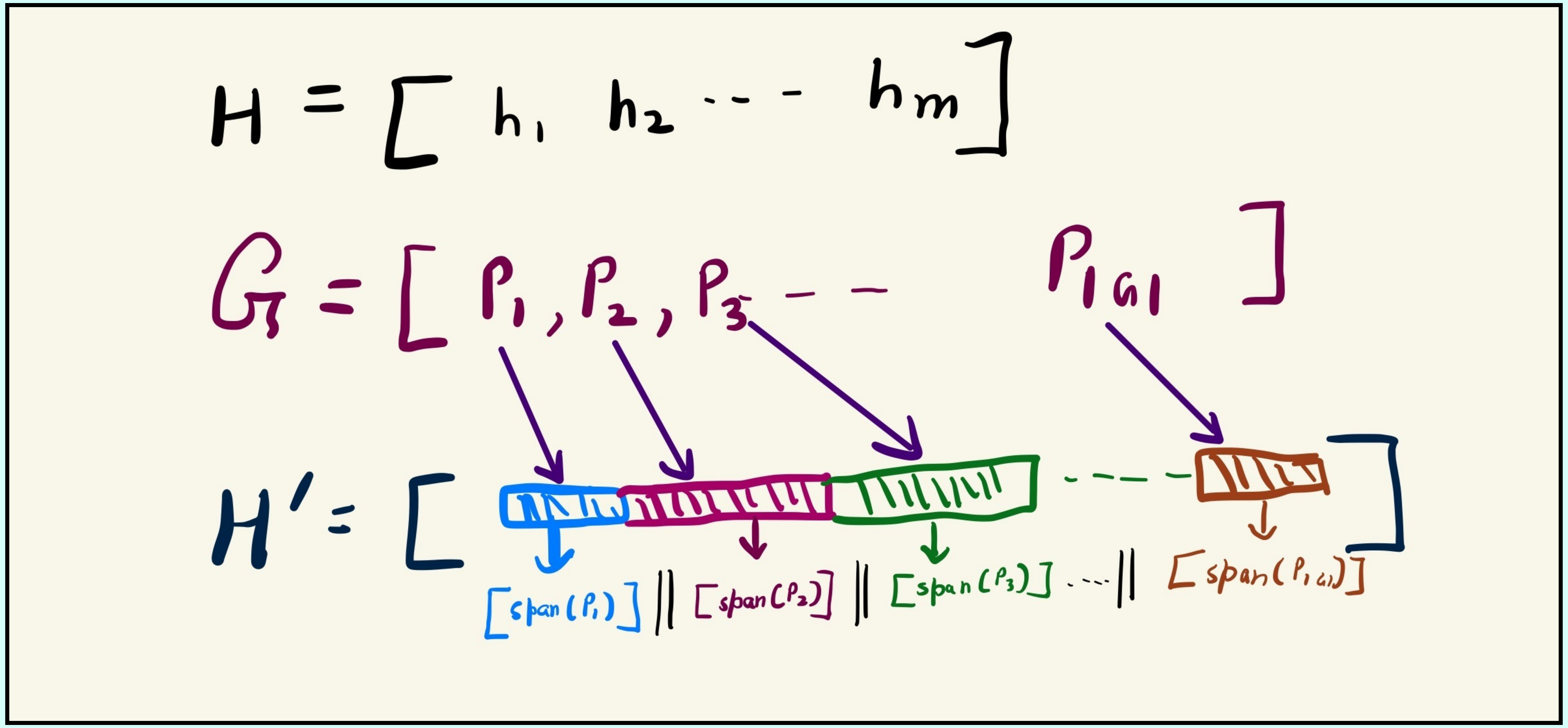
Naive Grouping Reduction



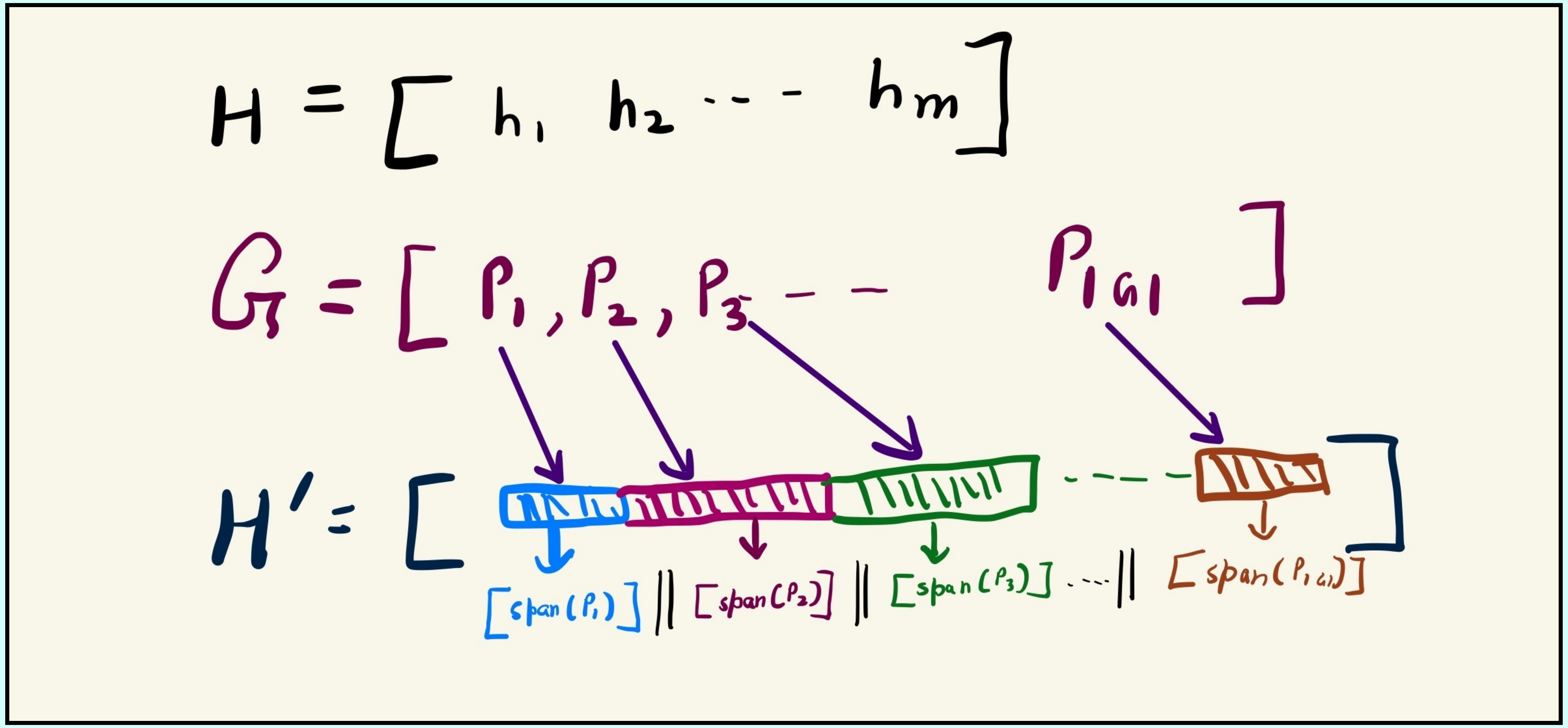
- Take G to be all collection of all subsets of size $\alpha m/k$ of $[m]$.



Naive Grouping Reduction



Naive Grouping Reduction

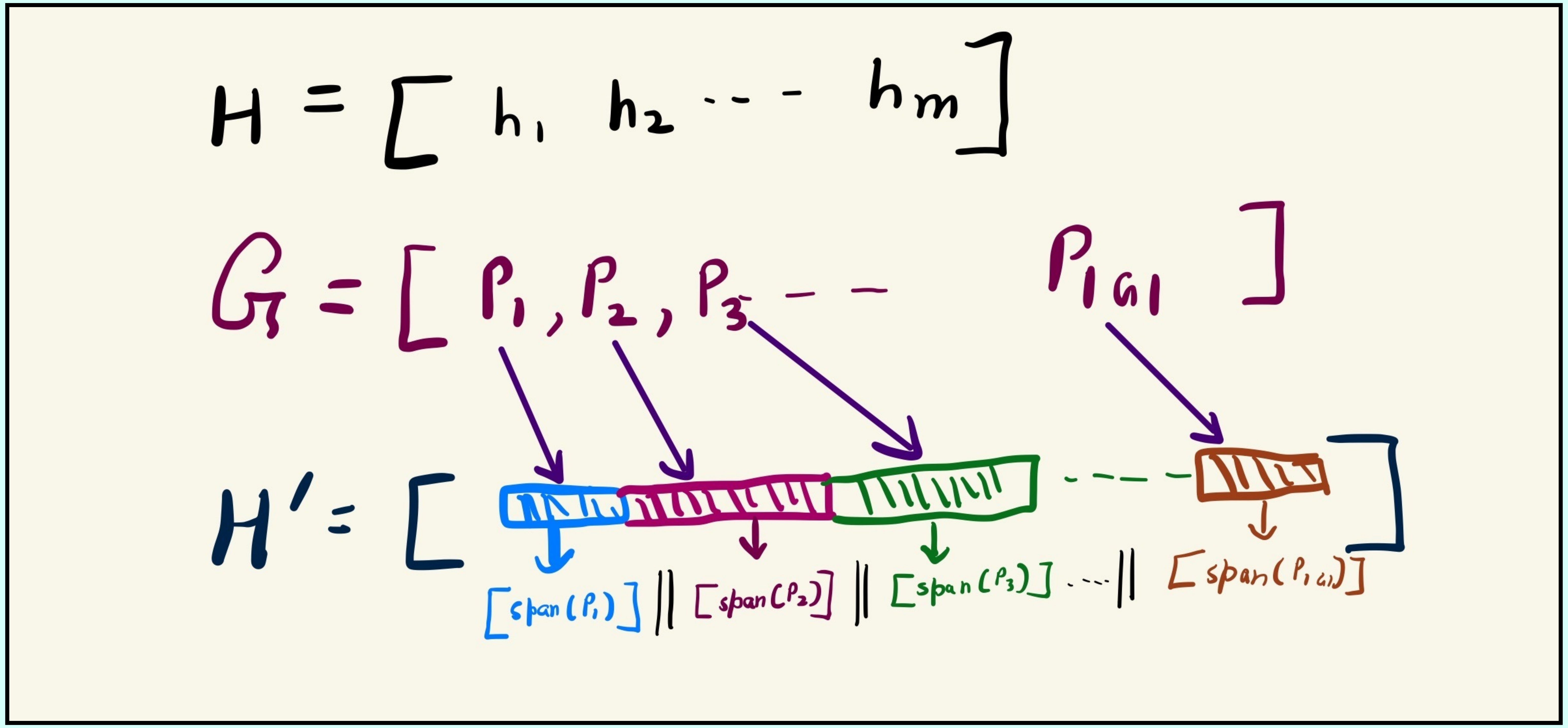


- New size,

$$M = q^{\alpha m/k} \cdot \binom{m}{\frac{\alpha m}{k}}$$

$$\leq 2^{O\left(\frac{m \log k}{k}\right)}$$

Naive Grouping Reduction

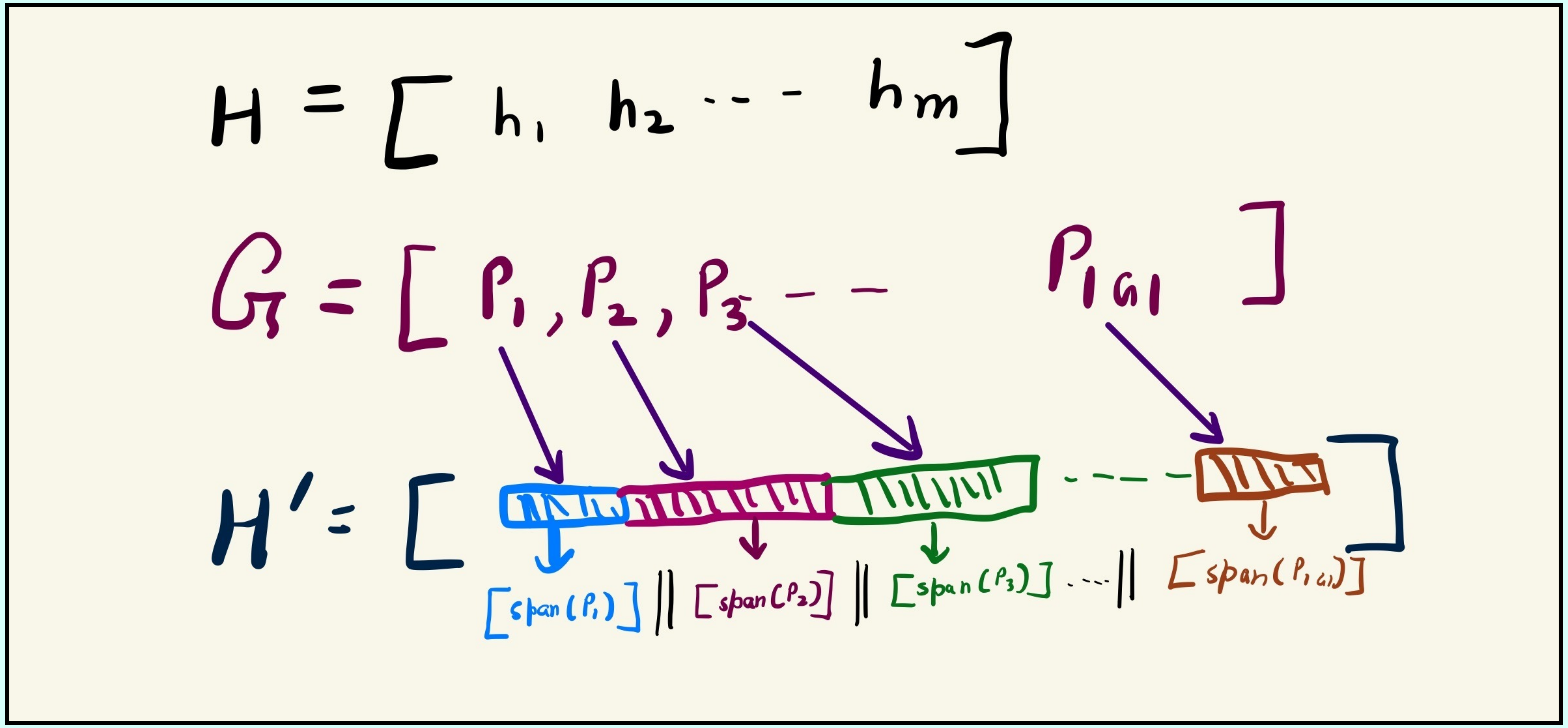


- New size,

$$M = q^{\alpha m/k} \cdot \binom{m}{\frac{\alpha m}{k}}$$

$$\leq 2^{O\left(\frac{m \log k}{k}\right)}$$

Naive **Grouping** Reduction



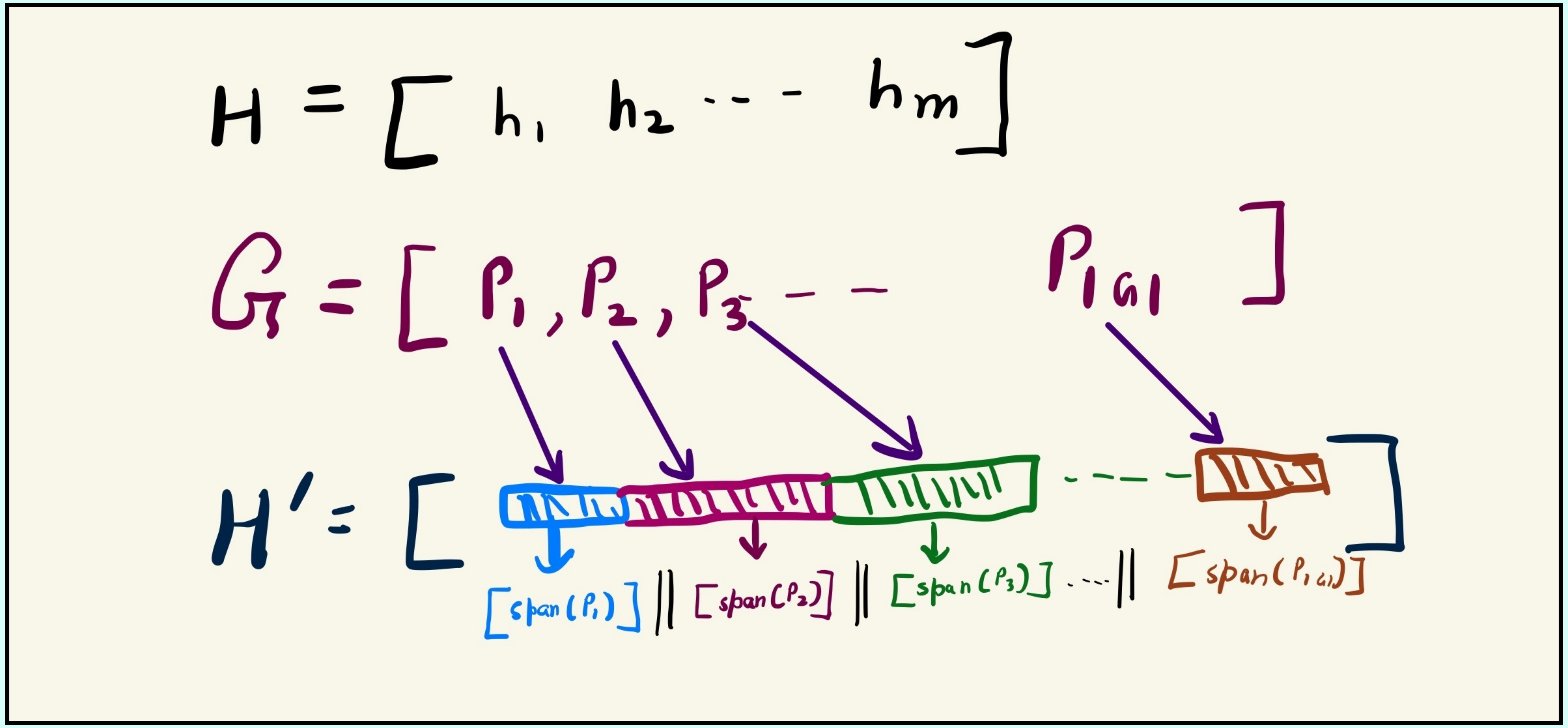
- New size,

$$M = q^{\alpha m/k} \cdot \binom{m}{\frac{\alpha m}{k}}$$

$$\leq 2^{O\left(\frac{m \log k}{k}\right)}$$

- Since $m = O(n_0)$, no $2^{o(n_0)}$ algorithm for **3-SAT**, implies no $M^{o(\frac{k}{\log k})}$ algorithm for **Gap-k-MLD**.

Naive **Grouping** Reduction



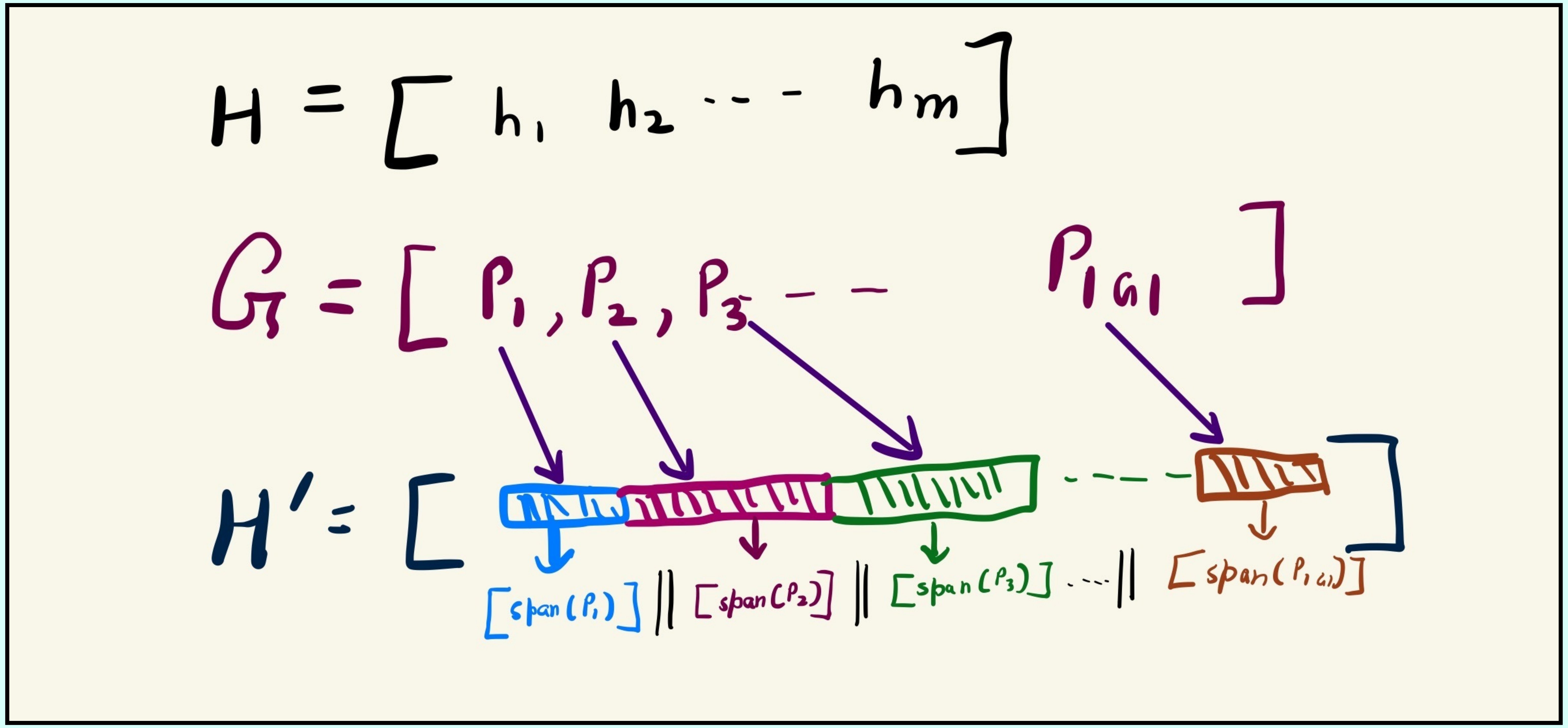
- New size,

$$M = q^{\alpha m/k} \cdot \binom{m}{\frac{\alpha m}{k}}$$

$$\leq 2^{O\left(\frac{m \log k}{k}\right)}$$

- Since $m = O(n_0)$, no $2^{o(n_0)}$ algorithm for **3-SAT**, implies no $M^{o(\frac{k}{\log k})}$ algorithm for **Gap-k-MLD**.

Naive **Grouping** Reduction



- New size,

$$M = q^{\alpha m/k} \cdot \binom{m}{\frac{\alpha m}{k}}$$

$$\leq 2^{O\left(\frac{m \log k}{k}\right)}$$

- Since $m = O(n_0)$, no $2^{o(n_0)}$ algorithm for **3-SAT**, implies no $M^{o(\frac{k}{\log k})}$ algorithm for **Gap-k-MLD**.

- Already better !!! 🥳

Cover Familly

Cover Family

$$G = \{ P_1, P_2 \text{ -------- }, P_{|G|} \}$$

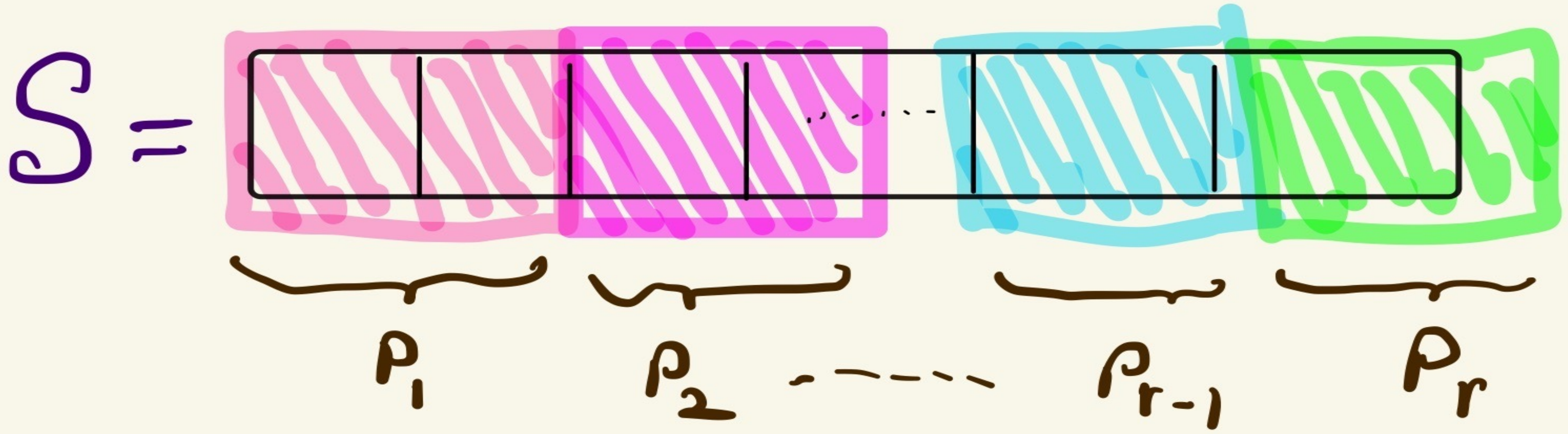
Some collection of subsets of $[m]$

- $P_i \subseteq [m]$
- $|P_i| \leq \frac{(1+\delta)\alpha m}{K}$
- $|G| \leq 2^{O(m/k)}$

We will use these subsets to group cleverly

Cover Family

$|S| = \alpha m$, $S = P_1 \sqcup P_2 \cdots \sqcup P_r$



$r \leq k$.

{Partition Property}

Cover Family

Cover Family

Cover Family

To group more cleverly we will need the following properties from the set of groups $G \subseteq 2^{[m]}$,

Cover Family

To group more cleverly we will need the following properties from the set of groups $G \subseteq 2^{[m]}$,

- **Partition Property:** For any subset S of αm elements of $[m]$, there exists $\leq k$ disjoint sets in G such that their union is exactly S.

Cover Family

To group more cleverly we will need the following properties from the set of groups $G \subseteq 2^{[m]}$,

- **Partition Property:** For any subset S of αm elements of $[m]$, there exists $\leq k$ disjoint sets in G such that their union is exactly S.
- **Size of Subsets:** Size of all subsets $P \in G$ to be bounded, $|P| \leq (1 + \delta)\alpha m / k$.

Cover Family

To group more cleverly we will need the following properties from the set of groups $G \subseteq 2^{[m]}$,

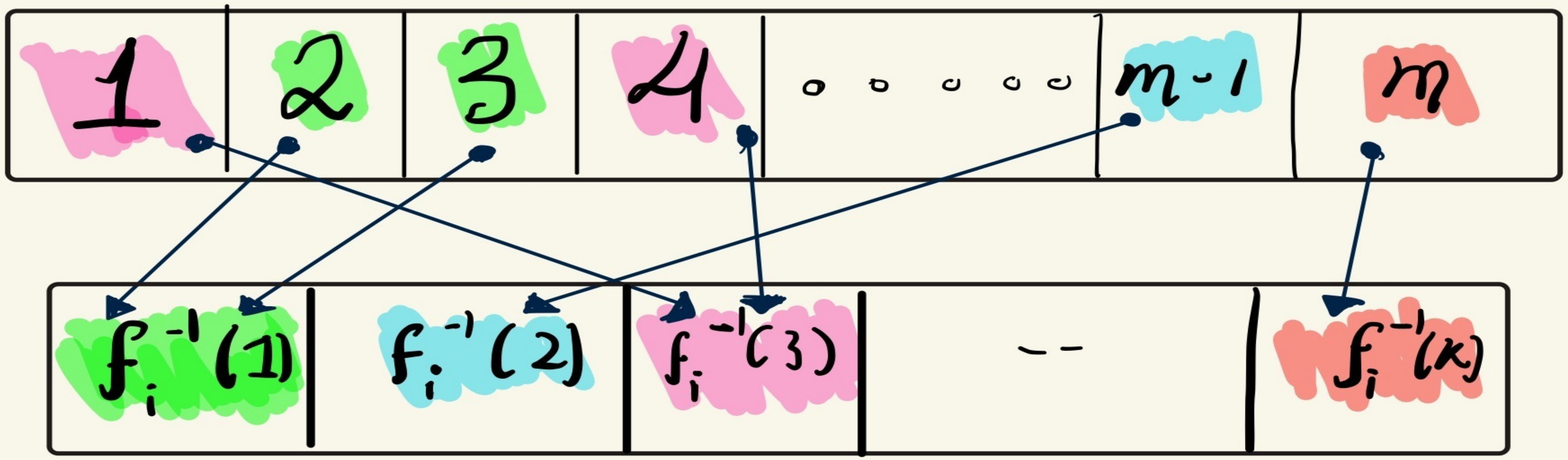
- **Partition Property:** For any subset S of αm elements of $[m]$, there exists $\leq k$ disjoint sets in G such that their union is exactly S.
- **Size of Subsets:** Size of all subsets $P \in G$ to be bounded, $|P| \leq (1+\delta)\alpha m/k$.
- **Size of Family:** I also want $|G| = 2^{O(m/k)}$.

Partition Family

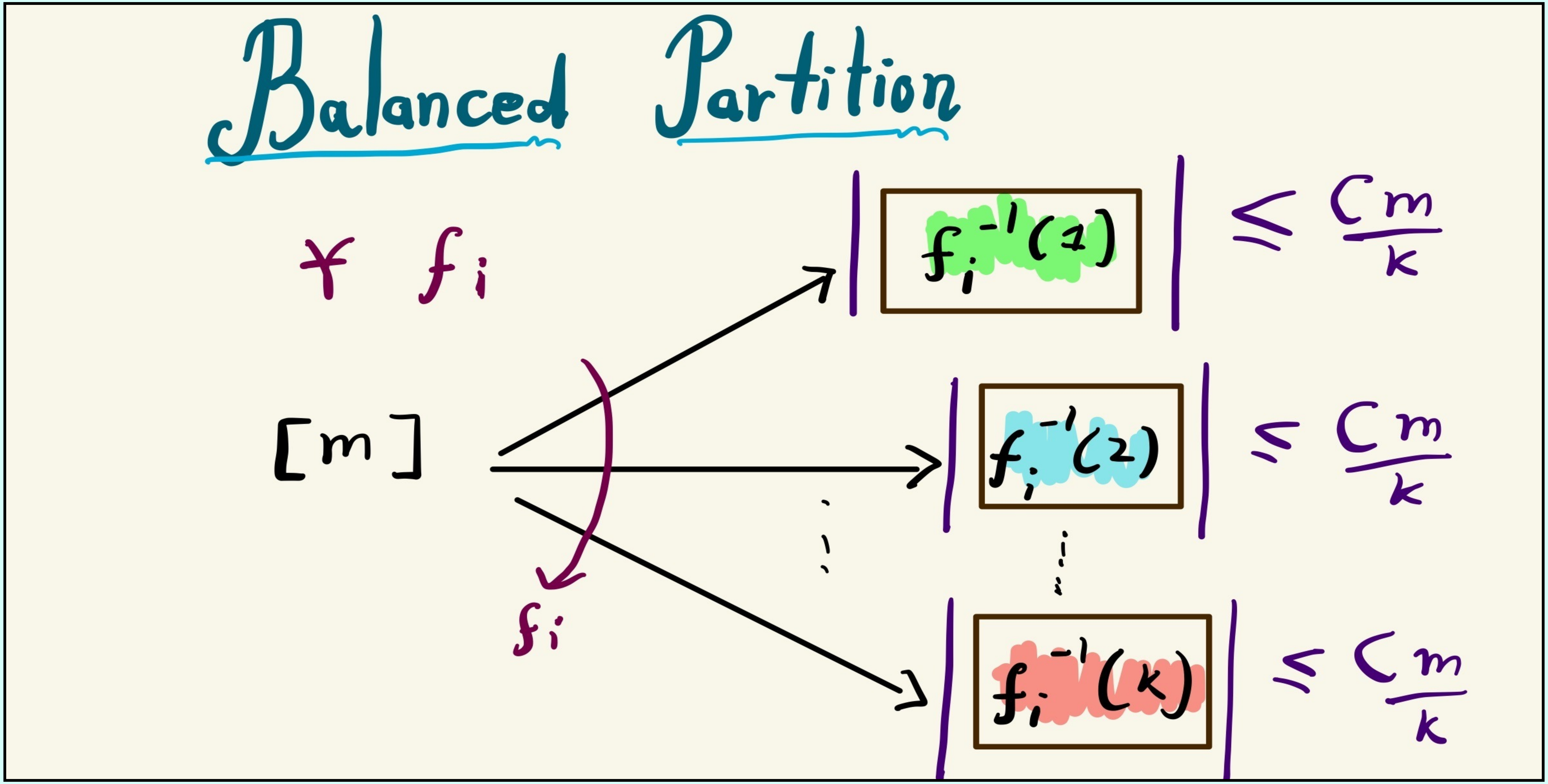
Partition Family

$$\mathcal{F} = \{ f_1, f_2, \ldots \ldots f_d \}$$

$$f_i : [m] \longrightarrow [k]$$



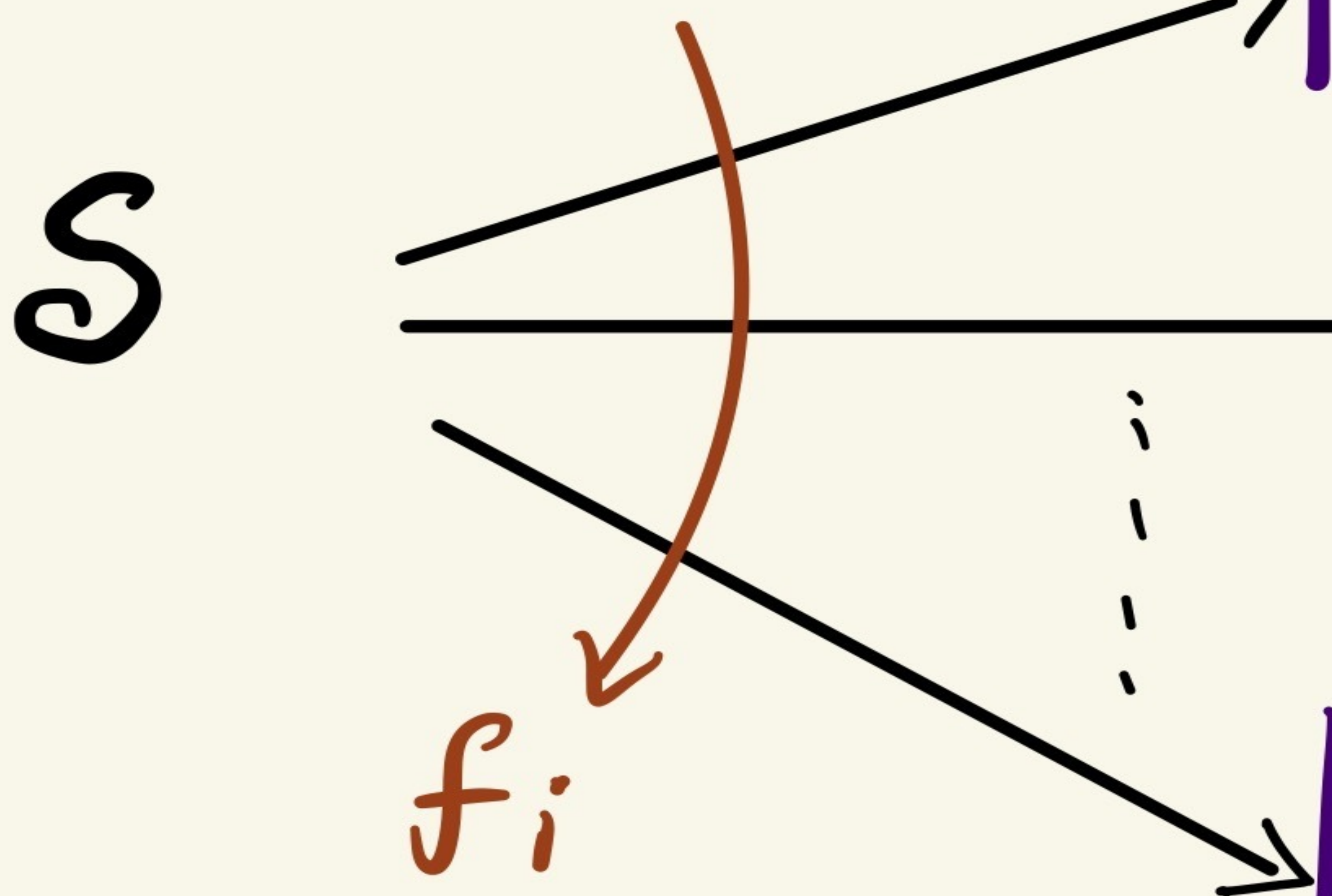
Partition Family



Partition Family

Subset Balancing

$\forall S, |S| = \alpha m, \exists f_i$



$$\left| \boxed{S \cap f_j^{-1}(1)} \right| < \frac{(1+\delta)\alpha m}{k}$$

$$\left| \boxed{S \cap f_j^{-1}(2)} \right| < \frac{(1+\delta)\alpha m}{k}$$

$$\vdots$$

$$\left| \boxed{S \cap f_j^{-1}(k)} \right| < \frac{(1+\delta)\alpha m}{k}$$

Partition Family

Partition Family

A set of partition functions $\{f_i\}_{i=1}^{d}$ from $[m]$ to $[k]$ such that,

Partition Family

A set of partition functions $\{f_i\}_{i=1}^{d}$ from $[m]$ to $[k]$ such that,

- **Balanced Partitions:** Every f_i is a balanced partition, meaning,

$$\forall j \in [k], |f_i^{-1}(j)| \leq Cm/k.$$

Partition Family

A set of partition functions $\{f_i\}_{i=1}^{d}$ from $[m]$ to $[k]$ such that,

- **Balanced Partitions:** Every f_i is a balanced partition, meaning,

$$\forall j \in [k], |f_i^{-1}(j)| \leq Cm/k.$$

- **Subset Balancing:** For every $S \subseteq [m]$ of size αm there exists a partition f_i such that f_i partitions S evenly, or in other words,

Partition Family

A set of partition functions $\{f_i\}_{i=1}^{d}$ from $[m]$ to $[k]$ such that,

- **Balanced Partitions:** Every f_i is a balanced partition, meaning,

$$\forall j \in [k], |f_i^{-1}(j)| \leq Cm/k.$$

- **Subset Balancing:** For every $S \subseteq [m]$ of size αm there exists a partition f_i such that f_i partitions S evenly, or in other words,

$$\forall j \in [k], |f_i^{-1}(j) \cap S| \leq (1+\delta)\alpha m/k.$$

Partition Family → Cover Family

Partition Family $\rightarrow$ Cover Family

- For every $f_i \in \{f_i\}_{i=1}^{d}$ and $j \in [k]$, consider $X_{ij} = f_i^{-1}(j)$.

Partition Family $\rightarrow$ Cover Family

- For every $f_i \in \{f_i\}_{i=1}^d$ and $j \in [k]$, consider $X_{ij} = f_i^{-1}(j)$.
- Now our cover family G will be just collection of all subsets of size $\leq (1+\delta)\alpha m/k$ of all X_{ij} .(**Size of Subsets**✅)

Partition Family $\rightarrow$ Cover Family

- For every $f_i \in \{f_i\}_{i=1}^{d}$ and $j \in [k]$, consider $X_{ij} = f_i^{-1}(j)$.
- Now our cover family G will be just collection of all subsets of size $\leq (1+\delta)\alpha m/k$ of all X_{ij} .(**Size of Subsets**✅)
- Since $|X_{ij}| \leq Cm/k$ implies number of subsets of X_{ij} is bounded by $2^{Cm/k}$.

Partition Family $\rightarrow$ Cover Family

- For every $f_i \in \{f_i\}_{i=1}^{d}$ and $j \in [k]$, consider $X_{ij} = f_i^{-1}(j)$.
- Now our cover family G will be just collection of all subsets of size $\leq (1+\delta)\alpha m/k$ of all X_{ij} .(**Size of Subsets**✅)
- Since $|X_{ij}| \leq Cm/k$ implies number of subsets of X_{ij} is bounded by $2^{Cm/k}$.
- Hence $|G| \leq 2^{Cm/k} \cdot d \cdot k$. (**Size of Family** ✅)

Partition Family $\to$ Cover Family

- For every $f_i \in \{f_i\}_{i=1}^{d}$ and $j \in [k]$, consider $X_{ij} = f_i^{-1}(j)$.
- Now our cover family G will be just collection of all subsets of size $\leq (1+\delta)\alpha m/k$ of all X_{ij} .(**Size of Subsets**✅)
- Since $|X_{ij}| \leq Cm/k$ implies number of subsets of X_{ij} is bounded by $2^{Cm/k}$.
- Hence $|G| \leq 2^{Cm/k} \cdot d \cdot k$. (**Size of Family** ✅)
- So we just need to bound d.

Partition Family $\rightarrow$ Cover Family

- For every $f_i \in \{f_i\}_{i=1}^{d}$ and $j \in [k]$, consider $X_{ij} = f_i^{-1}(j)$.
- Now our cover family G will be just collection of all subsets of size $\leq (1+\delta)\alpha m/k$ of all X_{ij} .(**Size of Subsets**✅)
- Since $|X_{ij}| \leq Cm/k$ implies number of subsets of X_{ij} is bounded by $2^{Cm/k}$.
- Hence $|G| \leq 2^{Cm/k} \cdot d \cdot k$. (**Size of Family** ✅)
- So we just need to bound d.
- We can show that if we sample f_i's uniformly at random the above can be attained with $d = O(k)$.

Partition Family → Cover Family

Partition Family → Cover Family

- Subset Balancing → Partition Property ✅

Partition Family → Cover Family

- **Subset Balancing** → **Partition Property** ✅

Subset Balancing: For every $S \subseteq [m]$ of size αm there exists a partition f_i such that f_i partitions S evenly, or in other words,

$$\forall j \in [k], |f_i^{-1}(j) \cap S| \leq (1+\delta)\alpha m/k.$$

↓

Partition Property: For any subset S of αm elements of $[m]$, there exists $\leq k$ disjoint sets in G such that their union is exactly S.

Partition Family → Cover Family

- Subset Balancing → Partition Property ✅

Subset Balancing: For every $S \subseteq [m]$ of size αm there exists a partition f_i such that f_i partitions S evenly, or in other words,

$$\forall j \in [k], |f_i^{-1}(j) \cap S| \leq (1+\delta)\alpha m/k.$$

Partition Property: For any subset S of αm elements of $[m]$, there exists $\leq k$ disjoint sets in G such that their union is exactly S.

- The $\leq k$ disjoint subsets are,

$$\sqcup_{j=1}^{k} \left(f_i^{-1}(j) \cap S\right)$$

- Since, $|f_i^{-1}(j) \cap S| \leq (1+\delta)\alpha m/k$
- This implies $\left(f_i^{-1}(j) \cap S\right) \in G$.

Derandomization

Derandomization of Partitions

$$
\begin{array}{cccccc}
 & 1 & 2 & \cdots & m \\
f_1 \to & g_1(1) & g_2(1) & \cdots & g_m(1) \\
f_2 \to & g_1(2) & g_2(2) & \cdots & g_m(2) \\
f_3 \to & g_1(3) & g_2(3) & \cdots & g_m(3) \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
f_d \to & g_1(d) & g_2(d) & \cdots & g_m(d)
\end{array}
$$

Derandomization of Partitions

- Set $d = \lceil \log_k m \rceil$.

$$
\begin{array}{cccccc}
 & 1 & 2 & \cdots & m \\
f_1 \rightarrow & g_1(1) & g_2(1) & \cdots & g_m(1) \\
f_2 \rightarrow & g_1(2) & g_2(2) & \cdots & g_m(2) \\
f_3 \rightarrow & g_1(3) & g_2(3) & \cdots & g_m(3) \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
f_d \rightarrow & g_1(d) & g_2(d) & \cdots & g_m(d)
\end{array}
$$

Derandomization of Partitions

- Set $d = \lceil \log_k m \rceil$.
- Now consider all functions from $[d] \to [k]$.

$$
\begin{array}{cccccc}
 & 1 & 2 & \cdots & m \\
f_1 \to & g_1(1) & g_2(1) & \cdots & g_m(1) \\
f_2 \to & g_1(2) & g_2(2) & \cdots & g_m(2) \\
f_3 \to & g_1(3) & g_2(3) & \cdots & g_m(3) \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
f_d \to & g_1(d) & g_2(d) & \cdots & g_m(d)
\end{array}
$$

Derandomization of Partitions

- Set $d = \lceil \log_k m \rceil$.
- Now consider all functions from $[d] \to [k]$.
- There will be at least m different functions $\{g_1, g_2 \cdots g_m\}$.

$$
\begin{array}{cccccc}
 & 1 & 2 & \cdots & m \\
f_1 \to & g_1(1) & g_2(1) & \cdots & g_m(1) \\
f_2 \to & g_1(2) & g_2(2) & \cdots & g_m(2) \\
f_3 \to & g_1(3) & g_2(3) & \cdots & g_m(3) \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
f_d \to & g_1(d) & g_2(d) & \cdots & g_m(d)
\end{array}
$$

Derandomization of Partitions

- Set $d = \lceil \log_k m \rceil$.
- Now consider all functions from $[d] \to [k]$.
- There will be at least m different functions $\{g_1, g_2 \cdots g_m\}$.
- Write the g_i's as the columns of a table and take partition functions f_i's to be the rows.

$$
\begin{array}{cccccc}
 & 1 & 2 & \cdots & m \\
f_1 \to & g_1(1) & g_2(1) & \cdots & g_m(1) \\
f_2 \to & g_1(2) & g_2(2) & \cdots & g_m(2) \\
f_3 \to & g_1(3) & g_2(3) & \cdots & g_m(3) \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
f_d \to & g_1(d) & g_2(d) & \cdots & g_m(d)
\end{array}
$$


Thank You!