Crowhammer

A key-recovery attack on **FALCON**

Calvin ABOU HAIDAR, Mehdi TIBOUCHI, Quentin PAYET

NTT

FALCON

Trivia

- Selected by **NIST** for **standardization** (FN-DSA)
- Efficient but complex (floating-point arithmetics, Gaussian sampler)

FALCON

Signature

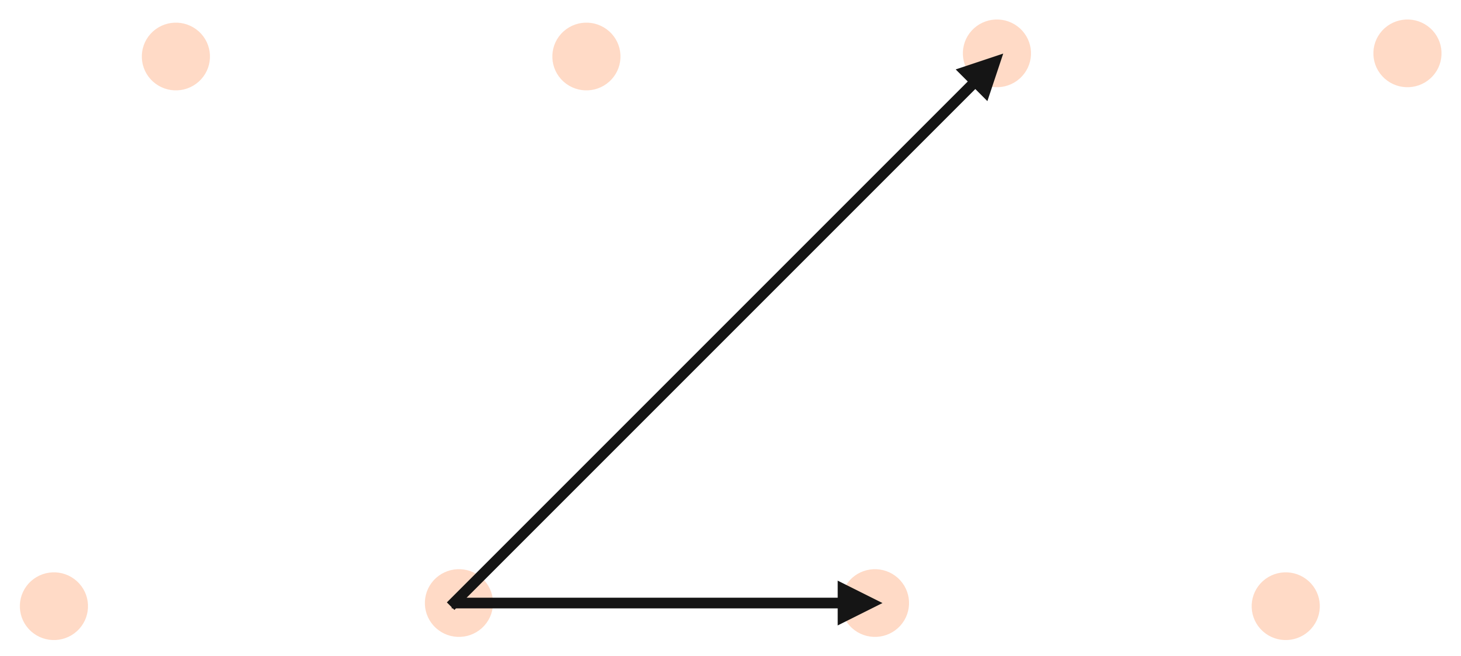
- Polynomial ring: $R = \mathbb{Z}_q[X]/(X^n + 1)$ modulo q
- NTRU Lattice: $L_h = \{(s_1, s_2) \in R^2 \mid s_2 + s_1 h = 0 \bmod q\}$ with $h = fg^{-1}$
- Secret key: small vectors $(g, -f) \in R^2, (G, -F) \in R^2$ and a "FALCON tree" T

$\begin{bmatrix} g & -f \\ G & -F \end{bmatrix}$ and $\begin{bmatrix} 1 & -h \\ 0 & q \end{bmatrix}$ are basis of L_h over R

FALCON

Signature

Nearest
Plane
Algorithm

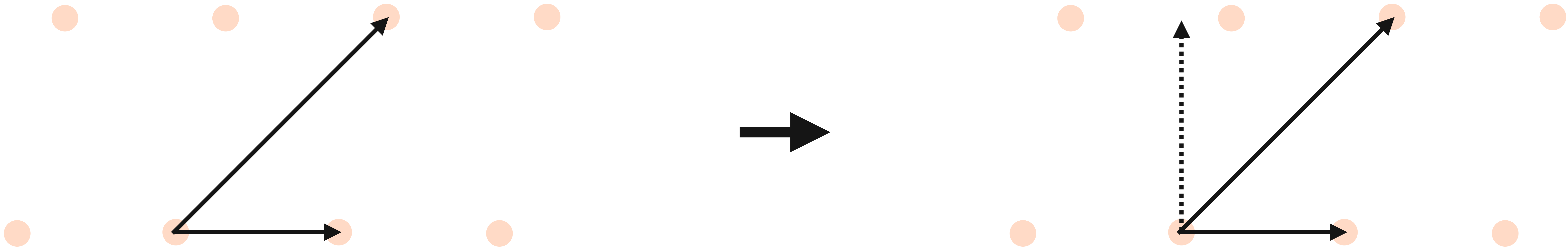


Original basis

FALCON

Signature

Nearest
Plane
Algorithm



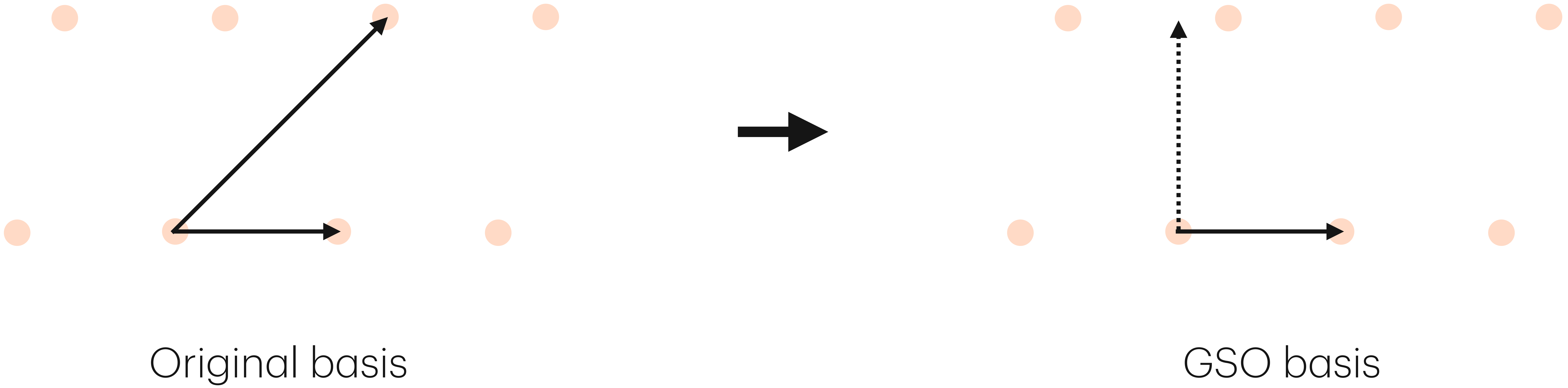
Original basis

GSO basis

FALCON

Signature

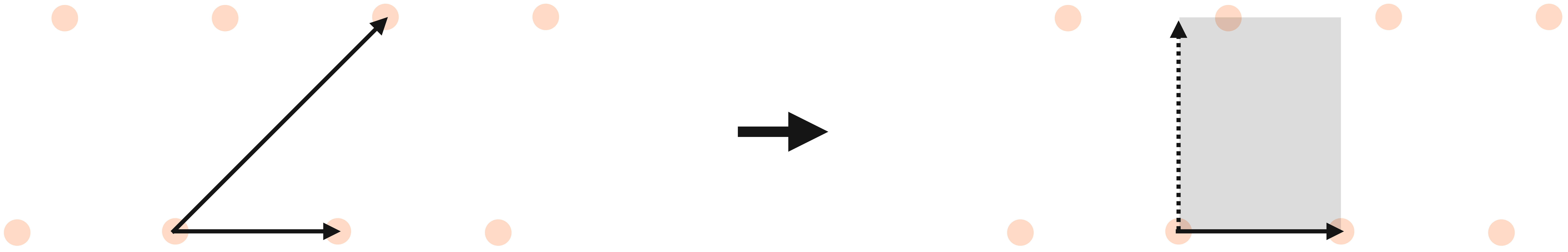
Nearest
Plane
Algorithm



FALCON

Signature

Nearest Plane Algorithm

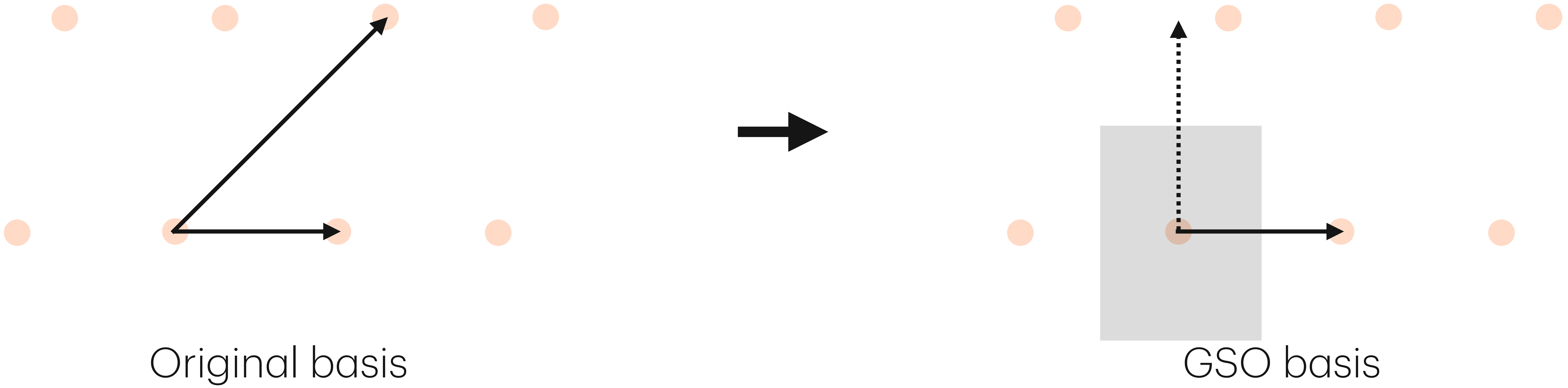


Original basis | GSO basis

FALCON

Signature

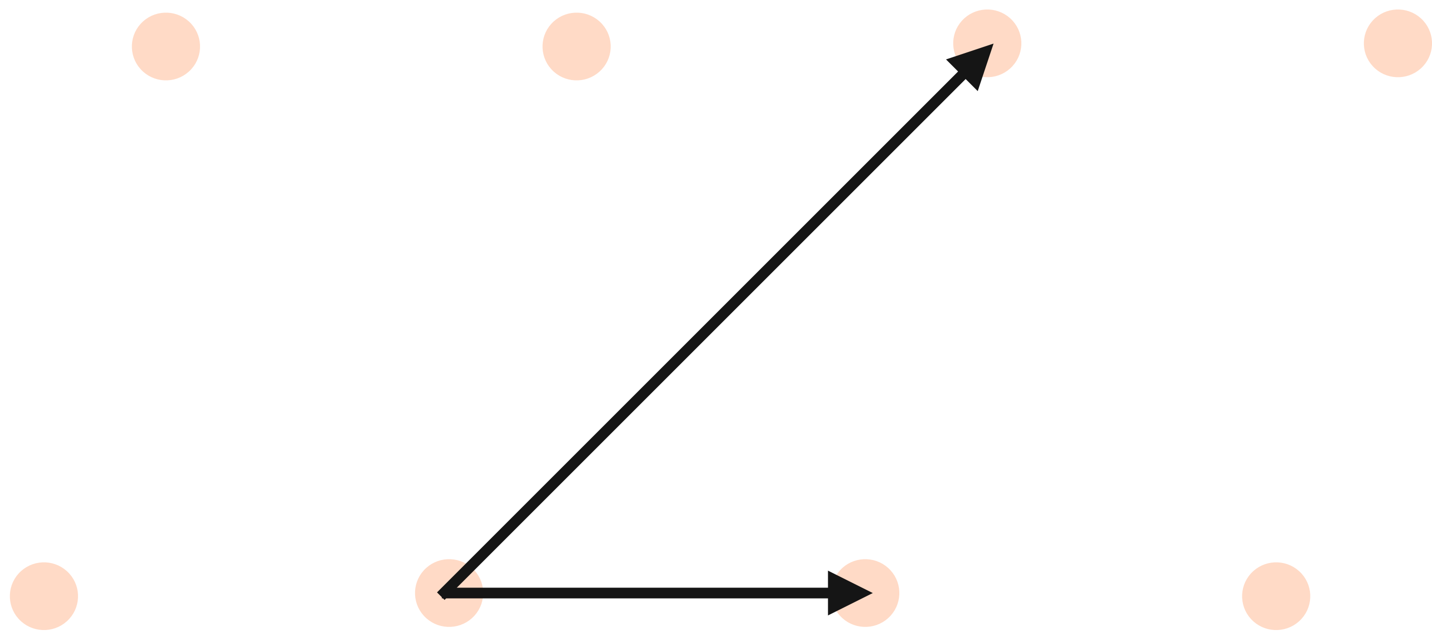
Nearest
Plane
Algorithm



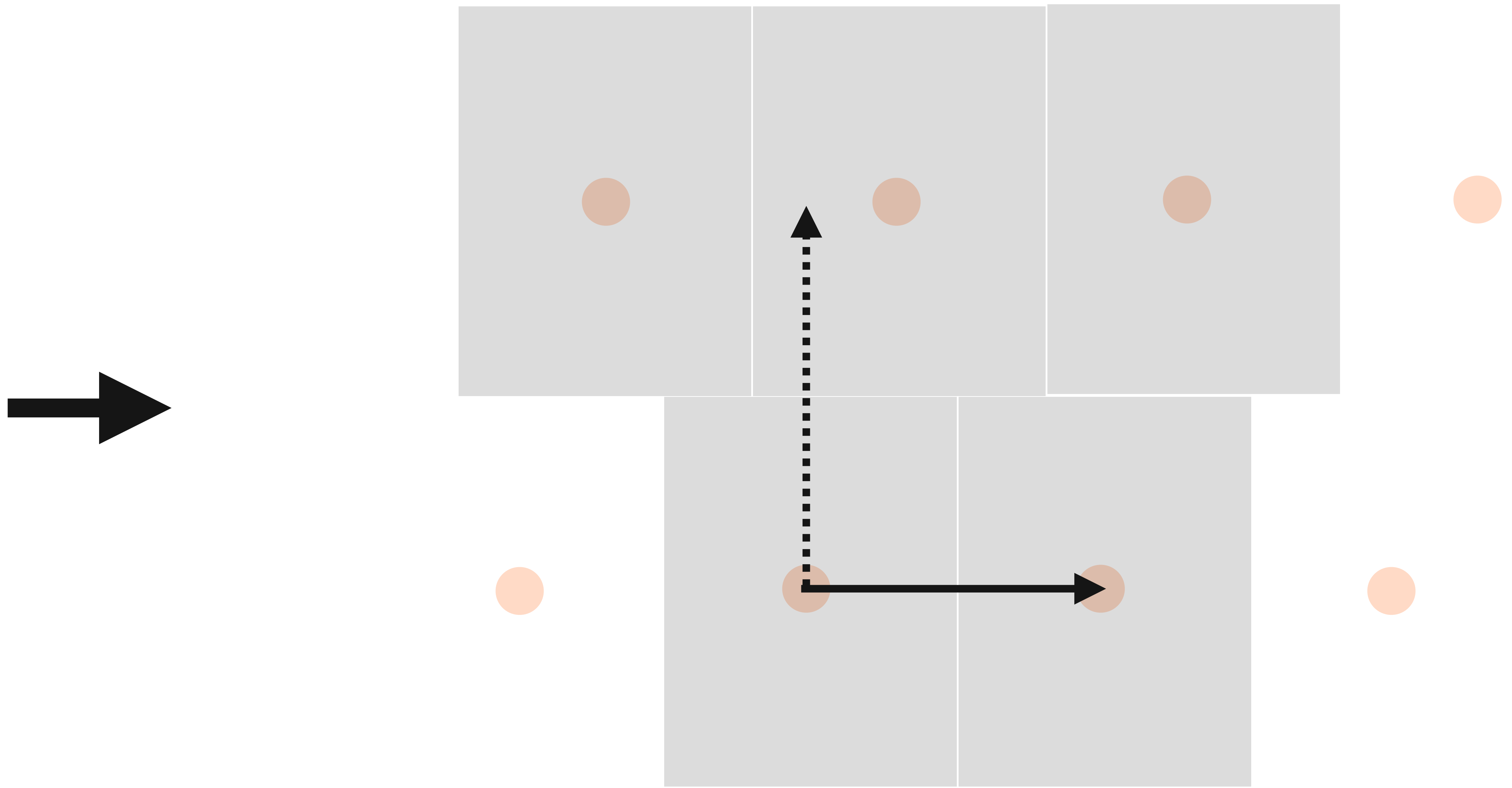
FALCON

Signature

Nearest
Plane
Algorithm



Original basis

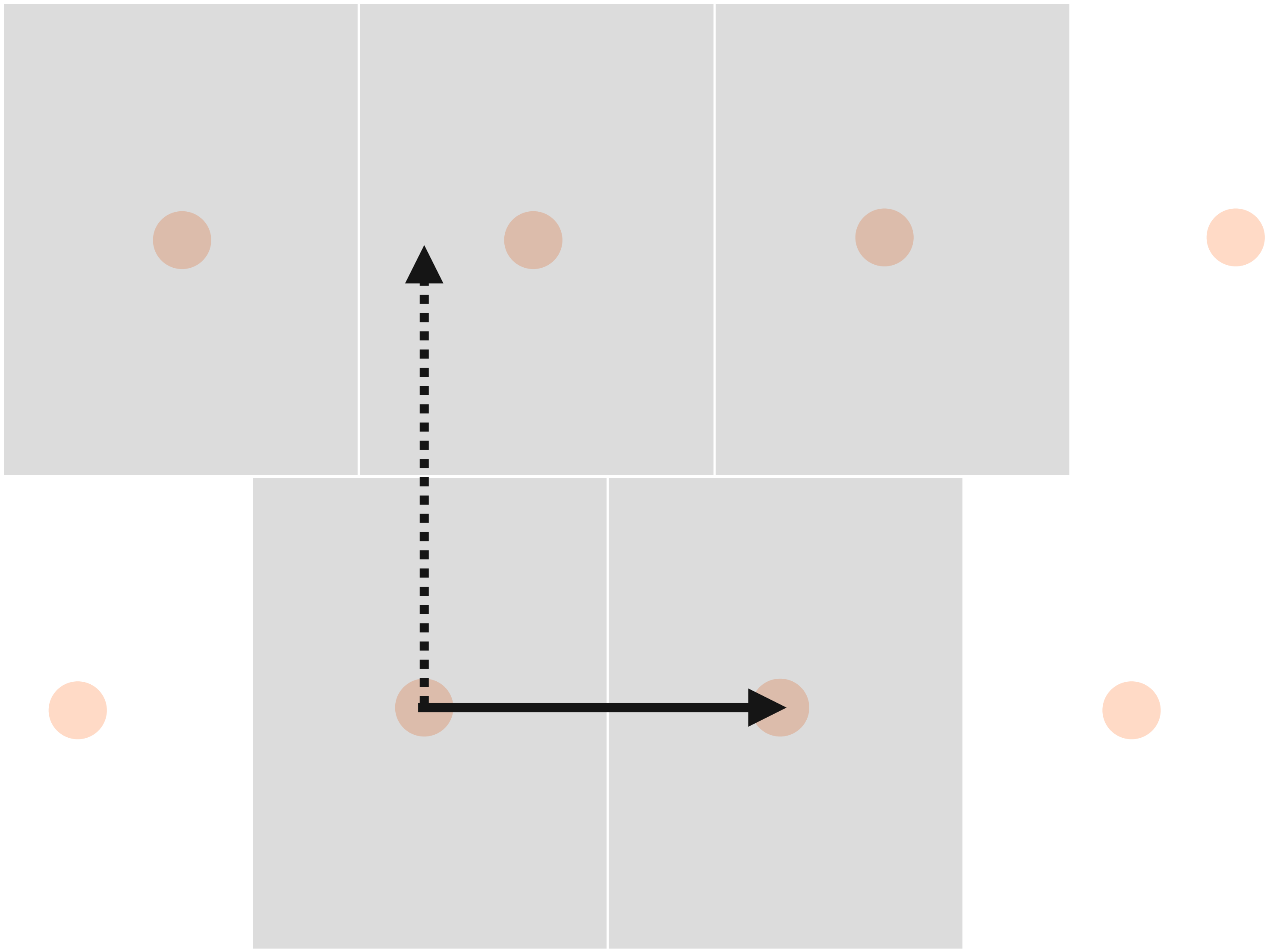


GSO basis

FALCON

Signature

Nearest Plane Algorithm



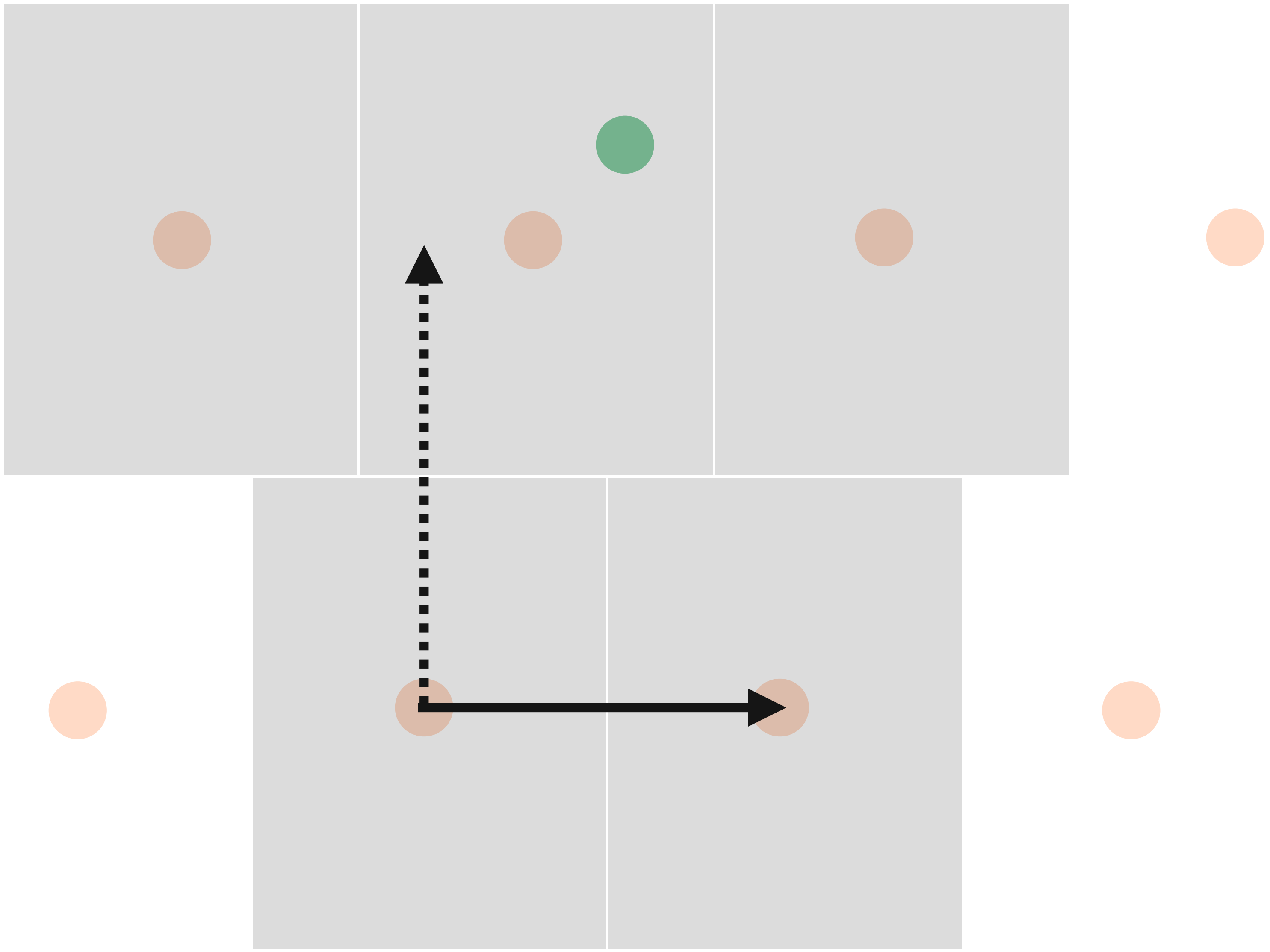
GSO basis

FALCON

Signature

Nearest
Plane
Algorithm

Approximate
Closest Vector Problem
Find ● "close" to ●



GSO basis

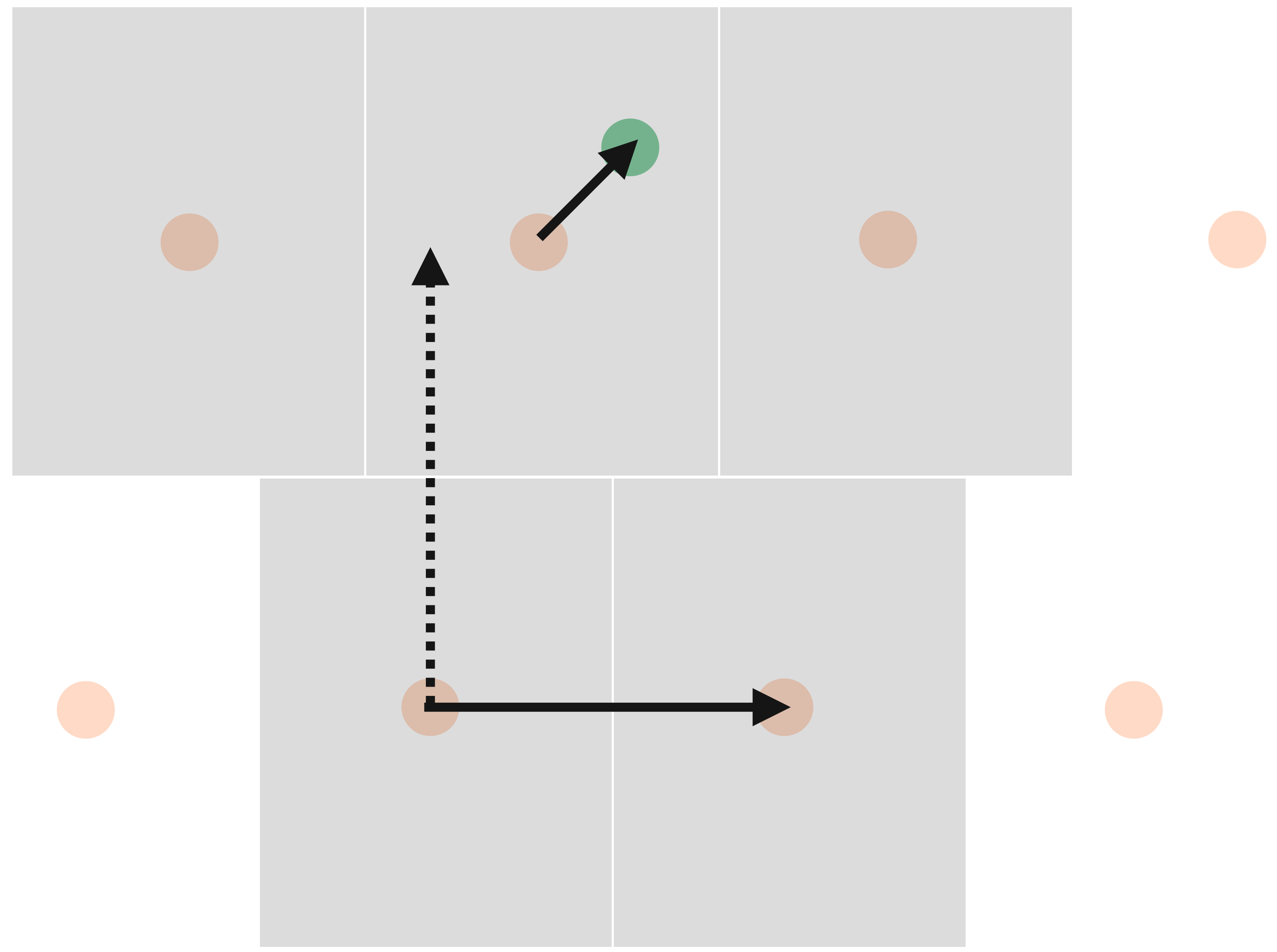
FALCON

Signature

Nearest Plane Algorithm

Approximate
Closest Vector Problem

Find ● "close" to ●

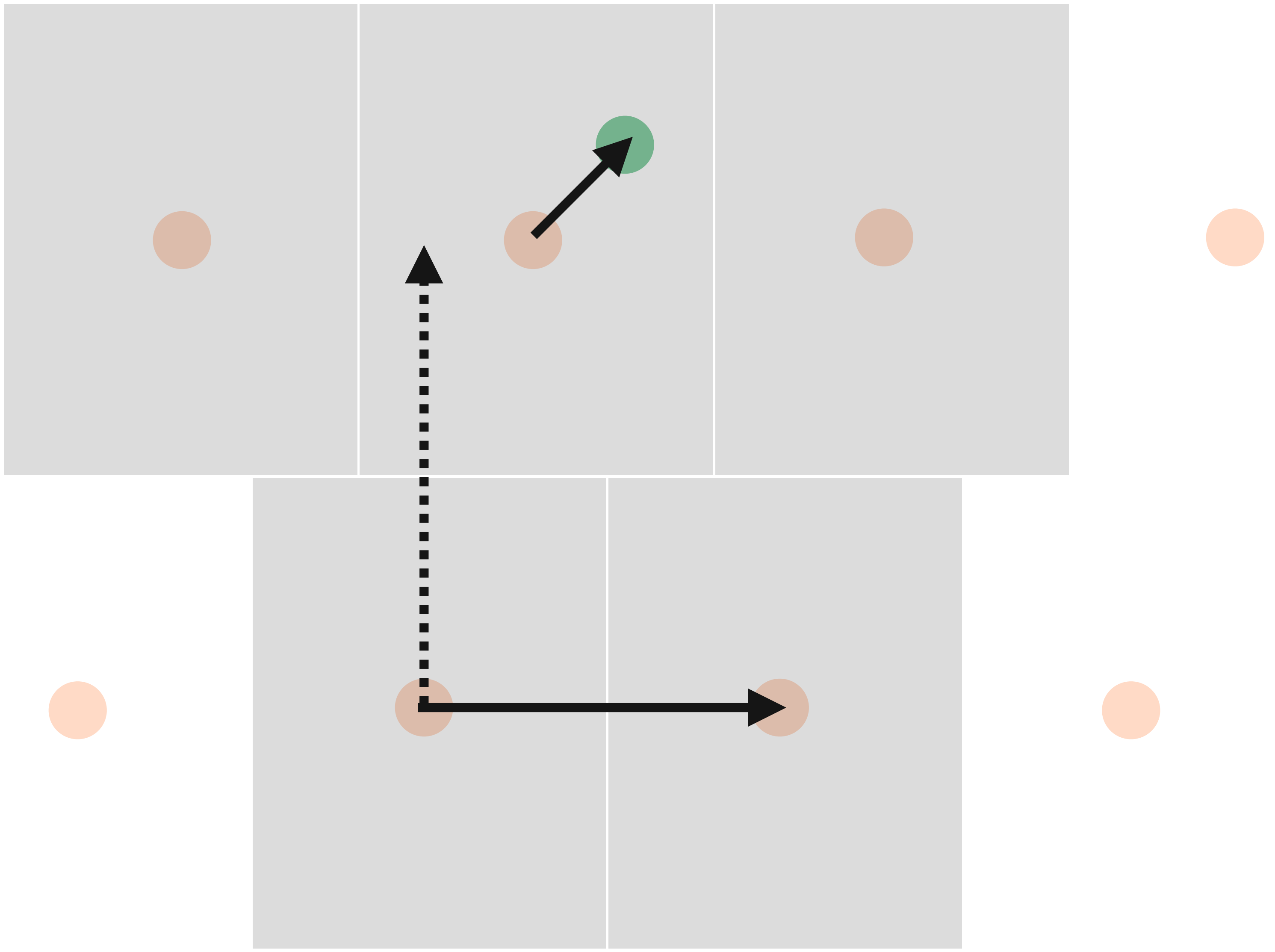


GSO basis

FALCON

Signature

Nearest
Plane
Algorithm



Approximate
Closest Vector Problem
Find ● "close" to ●

Broken by NR06

FALCON

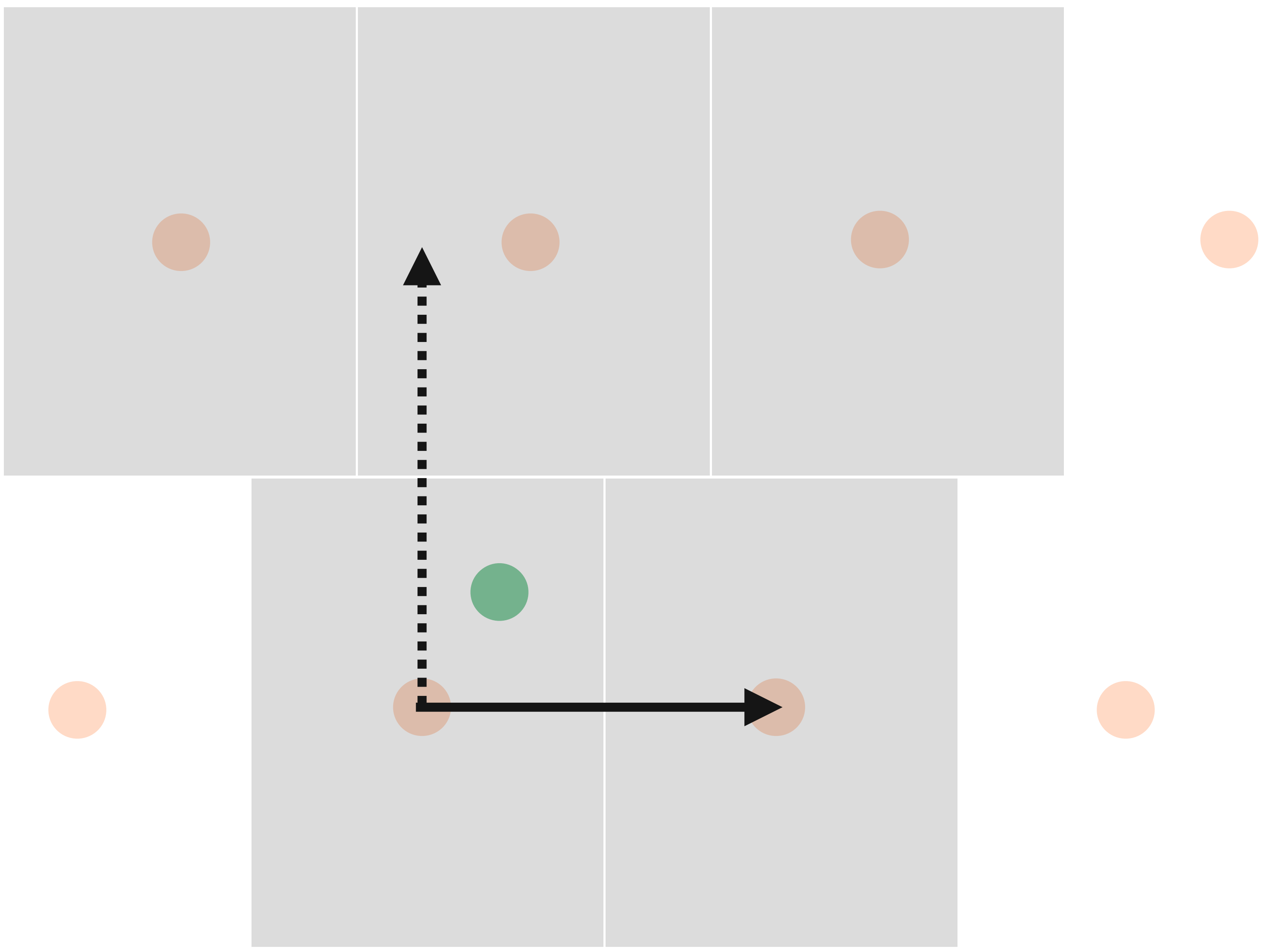
Signature

Nearest Plane Algorithm

Approximate

Closest Vector Problem

Find ● "close" to ●



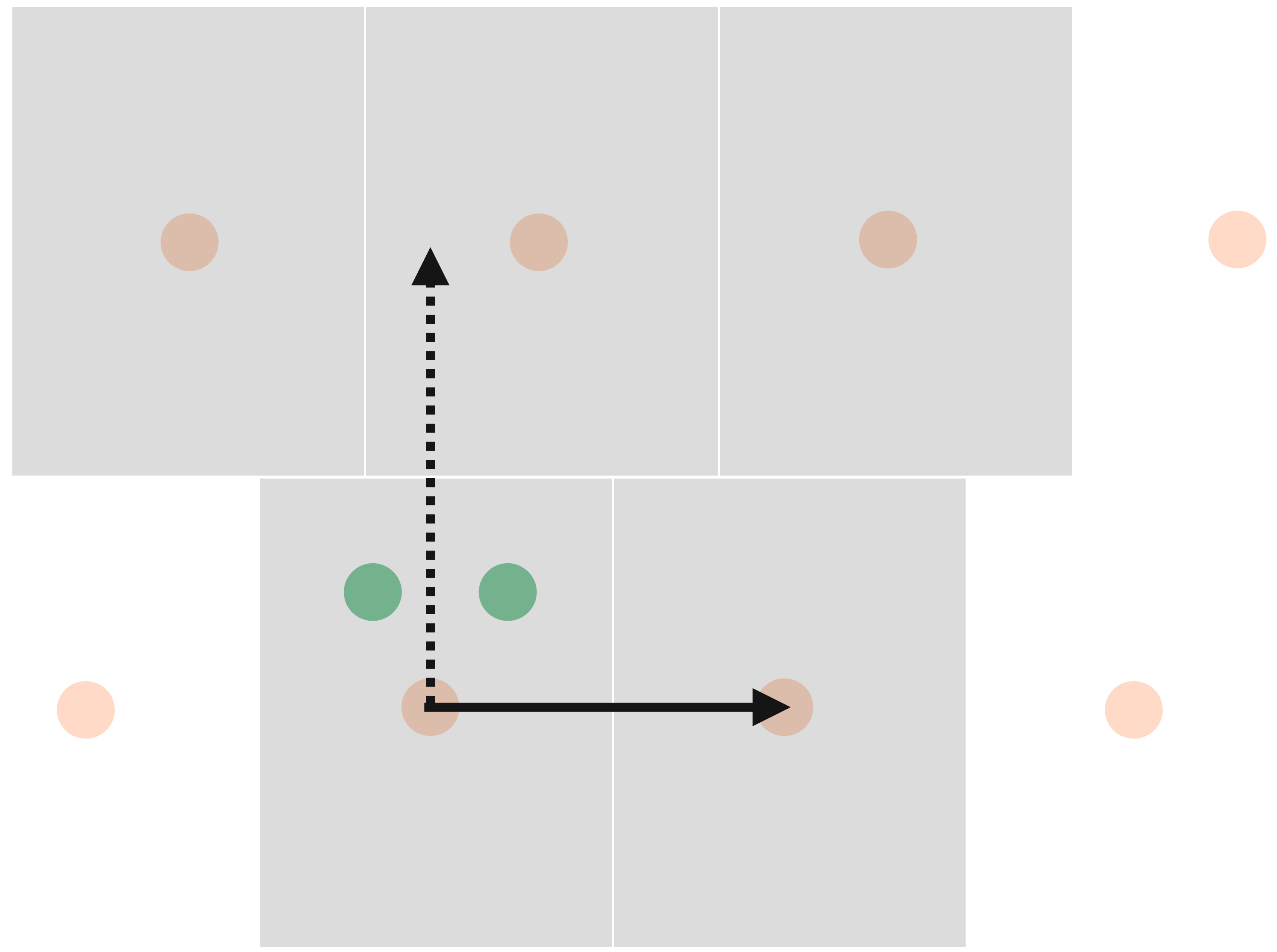
GSO basis

Broken by NR06

FALCON

Signature

Nearest
Plane
Algorithm



GSO basis

Approximate
Closest Vector Problem

Find ● "close" to ●

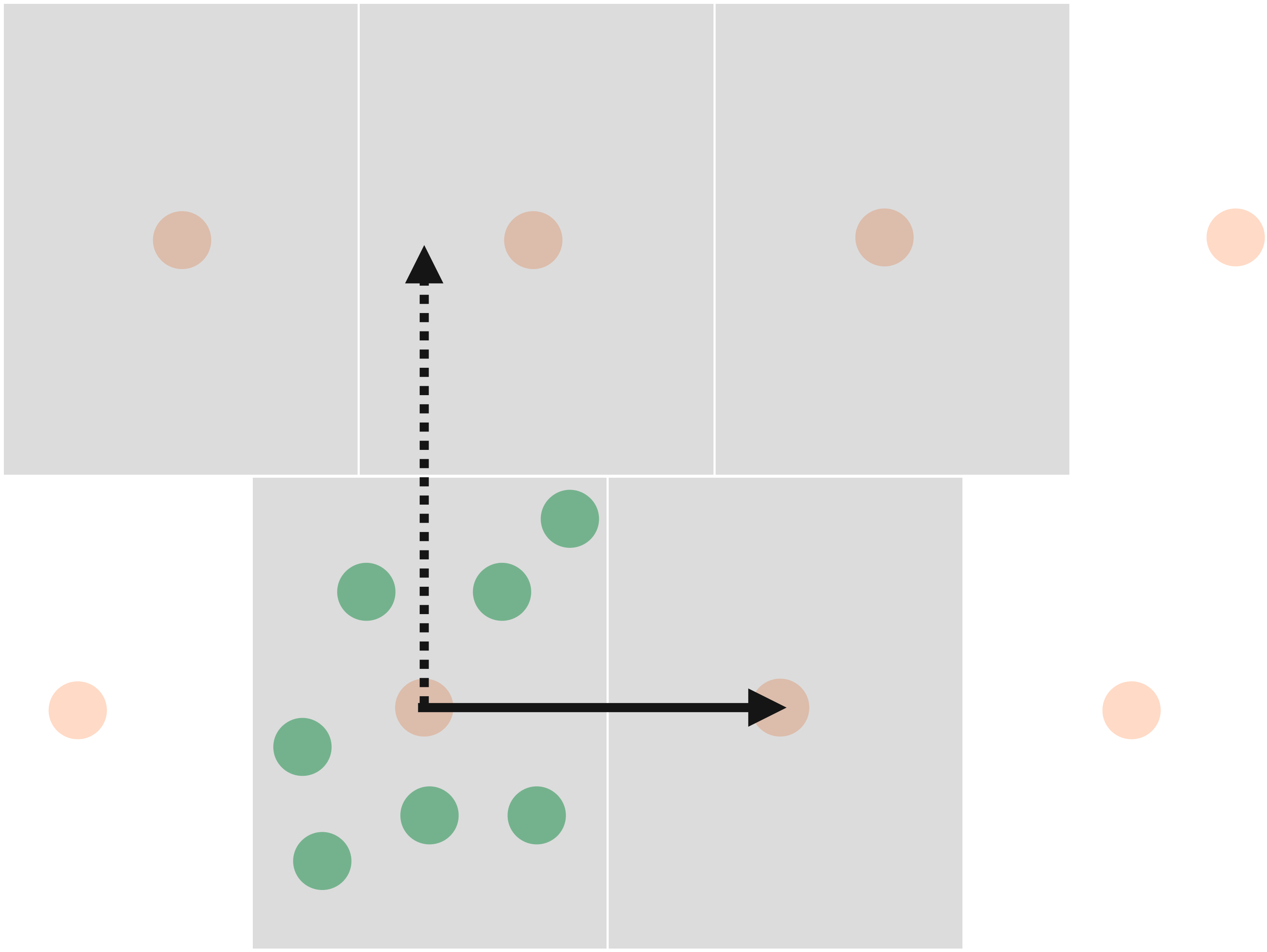
Broken by NR06

FALCON

Signature

Nearest
Plane
Algorithm

Approximate
Closest Vector Problem
Find ● "close" to ●



GSO basis

Broken by NR06

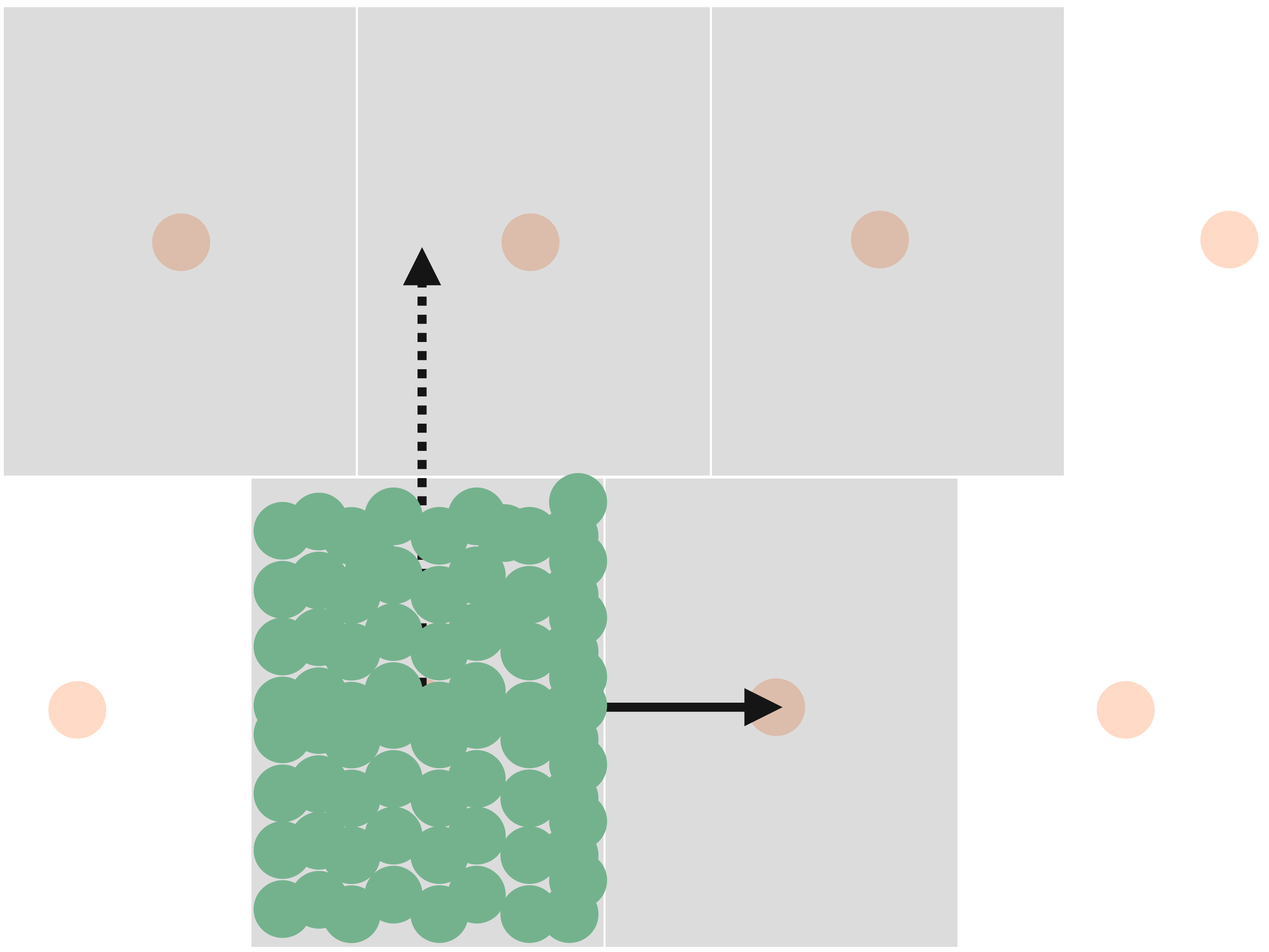
FALCON

Signature

Nearest Plane Algorithm

Approximate **Closest Vector Problem**

Find "close" to



GSO basis

Broken by NR06

FALCON

Distribution of signatures

FALCON signatures follow a discrete Gaussian distribution $\sim \mathscr{D}_{B^T \Sigma B}$

FALCON

Distribution of signatures

FALCON signatures follow a discrete Gaussian distribution $\sim \mathcal{D}_{B^T \Sigma B}$

We choose Σ to avoid leakage

FALCON

Distribution of signatures

FALCON signatures follow a discrete Gaussian distribution $\sim \mathscr{D}_{B^T \Sigma B}$

We choose Σ to avoid leakage

GSO decomposition $B = LD^{1/2}U$

FALCON

Distribution of signatures

FALCON signatures follow a discrete Gaussian distribution $\sim \mathcal{D}_{B^T\Sigma B}$

We choose Σ to avoid leakage

GSO decomposition $B = LD^{1/2}U$

If $\Sigma = L^{-T}\sigma^2 D^{-1}L^{-1}$ then $B^T\Sigma B = \sigma^2 Id$

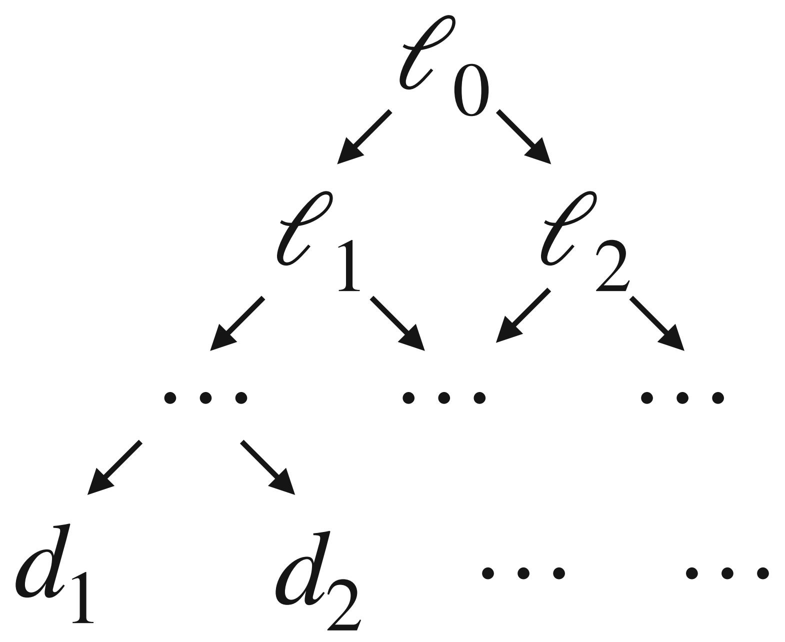
FALCON

Distribution of signatures

FALCON signatures follow a discrete Gaussian distribution $\sim \mathscr{D}_{B^T \Sigma B}$

If $\Sigma = L^{-T} \sigma^2 D^{-1} L^{-1}$ then $B^T \Sigma B = \sigma^2 Id$

FALCON Tree



to sample Σ

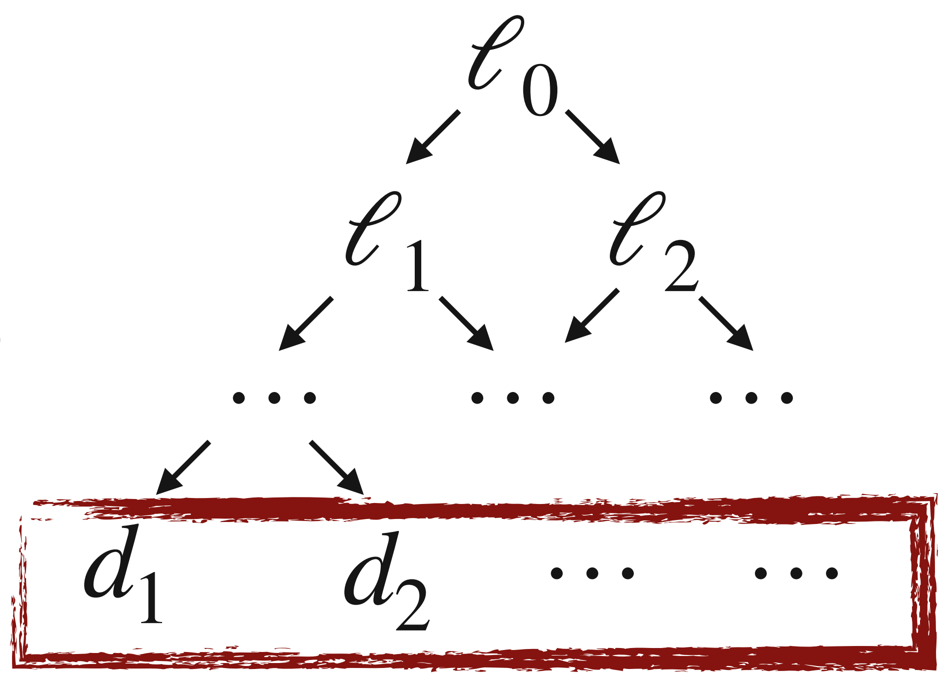
FALCON

Distribution of signatures

FALCON signatures follow a discrete Gaussian distribution $\sim \mathscr{D}_{B^T \Sigma B}$

If $\Sigma = L^{-T} \sigma^2 D^{-1} L^{-1}$ then $B^T \Sigma B = \sigma^2 Id$

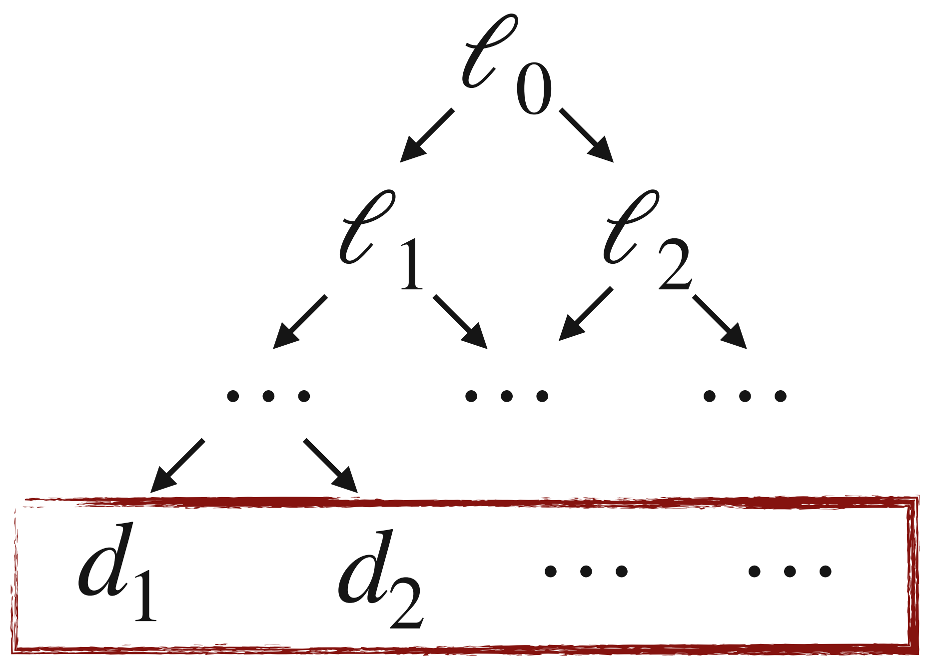
FALCON Tree



to sample Σ

FALCON

Gaussian Sampling

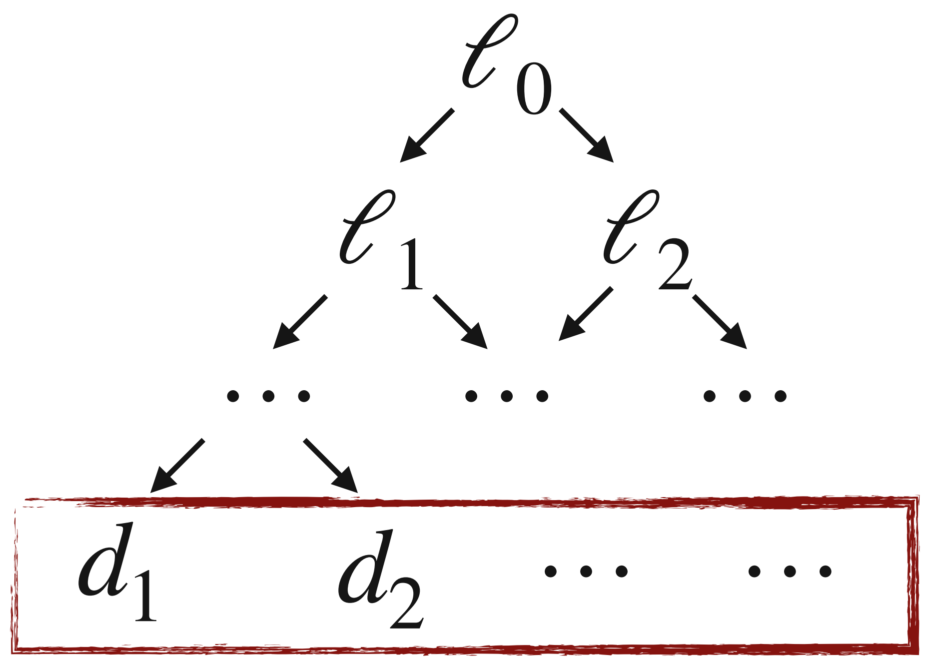


Strategy: Perform (half) Gaussian sampling over $\mathbb{Z}$ then merge

Implementation: constant-time, linear scan over a table

FALCON

Gaussian Sampling



Strategy: Perform (half) Gaussian sampling over $\mathbb{Z}$ then merge

Implementation: constant-time, linear scan over a table

```
def sampleZ():
    u = sampleUniform(72)
    s = 0
    for i in range(N):
        if RCDT[i] > u:
            s += 1
    return s
```

FALCON

Gaussian Sampling

Smaller `RCDT[*]` $\Longrightarrow$ **Smaller Gaussian**

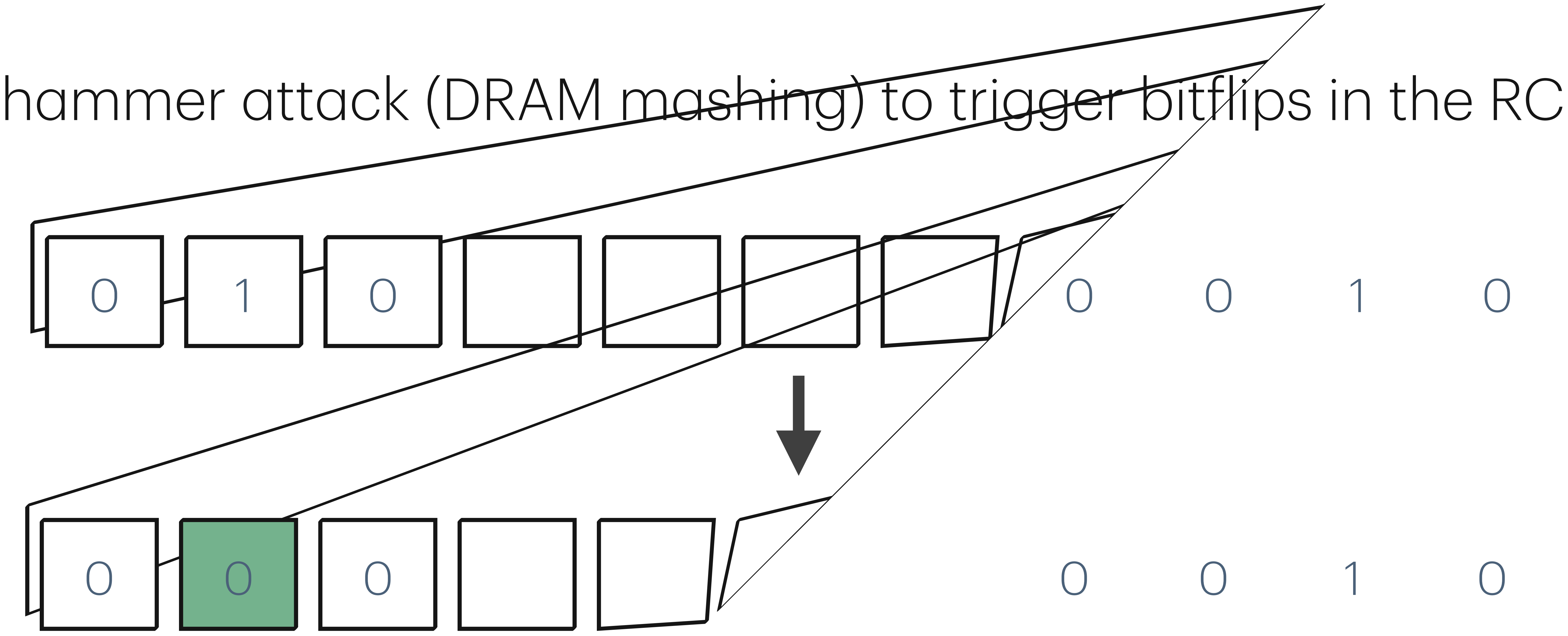
```
def sampleZ():
    u = sampleUniform()
    s = 0
    for i in range(N):
        if RCDT[i] > u:
            s += 1
    return s
```

Rowhammer

Attacking DRAM

Idea: Attack `RCDT[*]` to lower its values and cause statistical leakage

Tool: Rowhammer attack (DRAM mashing) to trigger bitflips in the RCDT



Attack

Nguyen-Regev

How many bitflips to work?

	Nguyen-Regev Attack
Full flip (Empty table)	✔
8 bitflips	✘
1 bitflip	✘

Not realistic

Scenario of Fahr et al. on FrodoKEM (CCS2022)

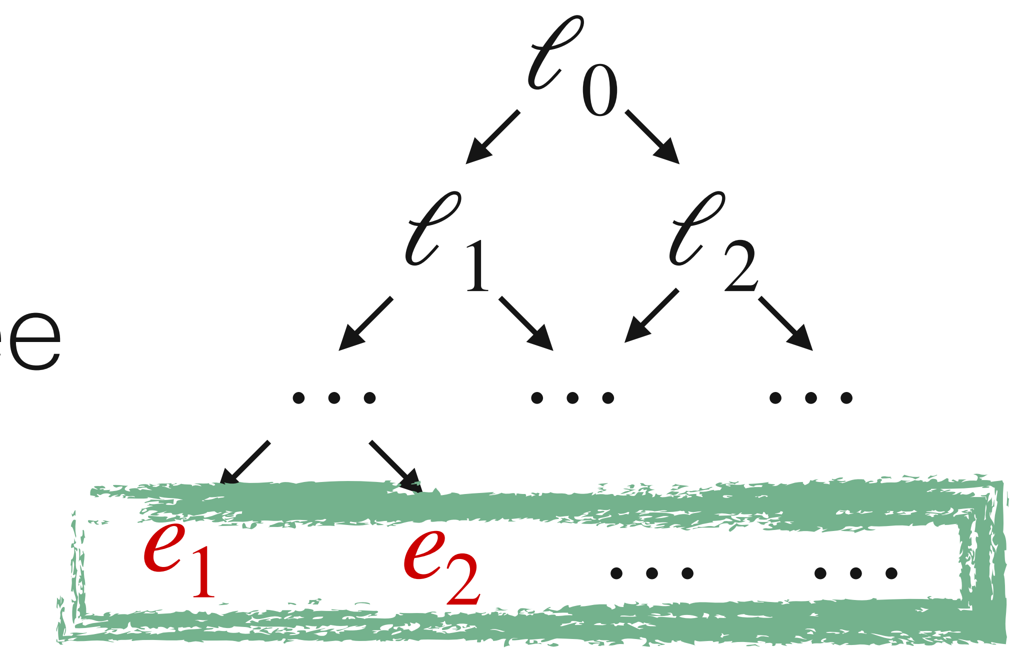
Attack

Effects of bitflips

Question: How does the signature distribution behave with bitflips?

If $\Sigma = L^{-T}\sigma^2 D^{-1} L^{-1}$ then $B^T \Sigma B = \sigma^2 Id$

FALCON Tree



to sample Σ

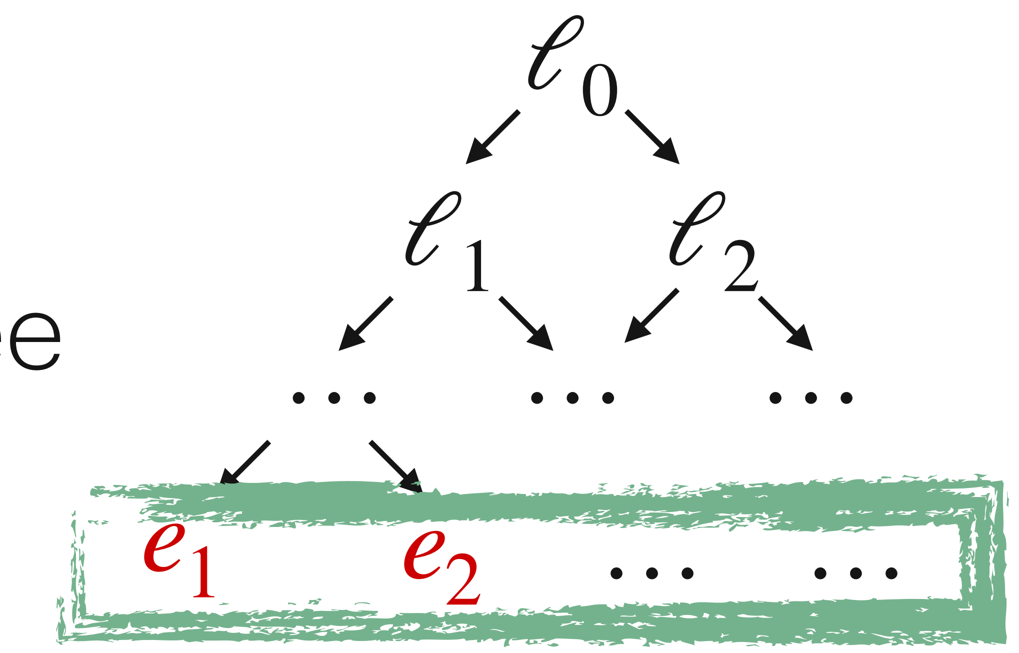
Attack

Effects of bitflips

Question: How does the signature distribution behave with bitflips?

$$\text{If } \Sigma = L^{-T}\sigma^2 E^{-1} L^{-1} \text{ then } B^T \Sigma B = \sigma^2 U^T D E^{-1} U = \tilde{\Sigma}$$

FALCON Tree



to sample Σ

Attack

Effects of bitflips

Question: How does the signature distribution behave with bitflips?

$$\text{If } \Sigma = L^{-T}E^{-1}L^{-1} \text{ then } B^T\Sigma B = \sigma^2 U^T D E^{-1} U = \tilde{\Sigma}$$

Observation: the vectors of the normalized GSO U are eigenvectors of $\tilde{\Sigma}$

Attack

Effects of bitflips

Question: How does the signature distribution behave with bitflips?

If $\Sigma = L^{-T}E^{-1}L^{-1}$ then $B^T\Sigma B = \sigma^2 U^T D E^{-1} U = \tilde{\Sigma}$

Observation: the vectors of the normalized GSO U are eigenvectors of $\tilde{\Sigma}$

Idea: get a good approximation of $\tilde{\Sigma}$ and compute its eigenvectors

Attack

Eigenvalue attack

Idea: get a good approximation of $\tilde{\Sigma}$ and compute its eigenvectors

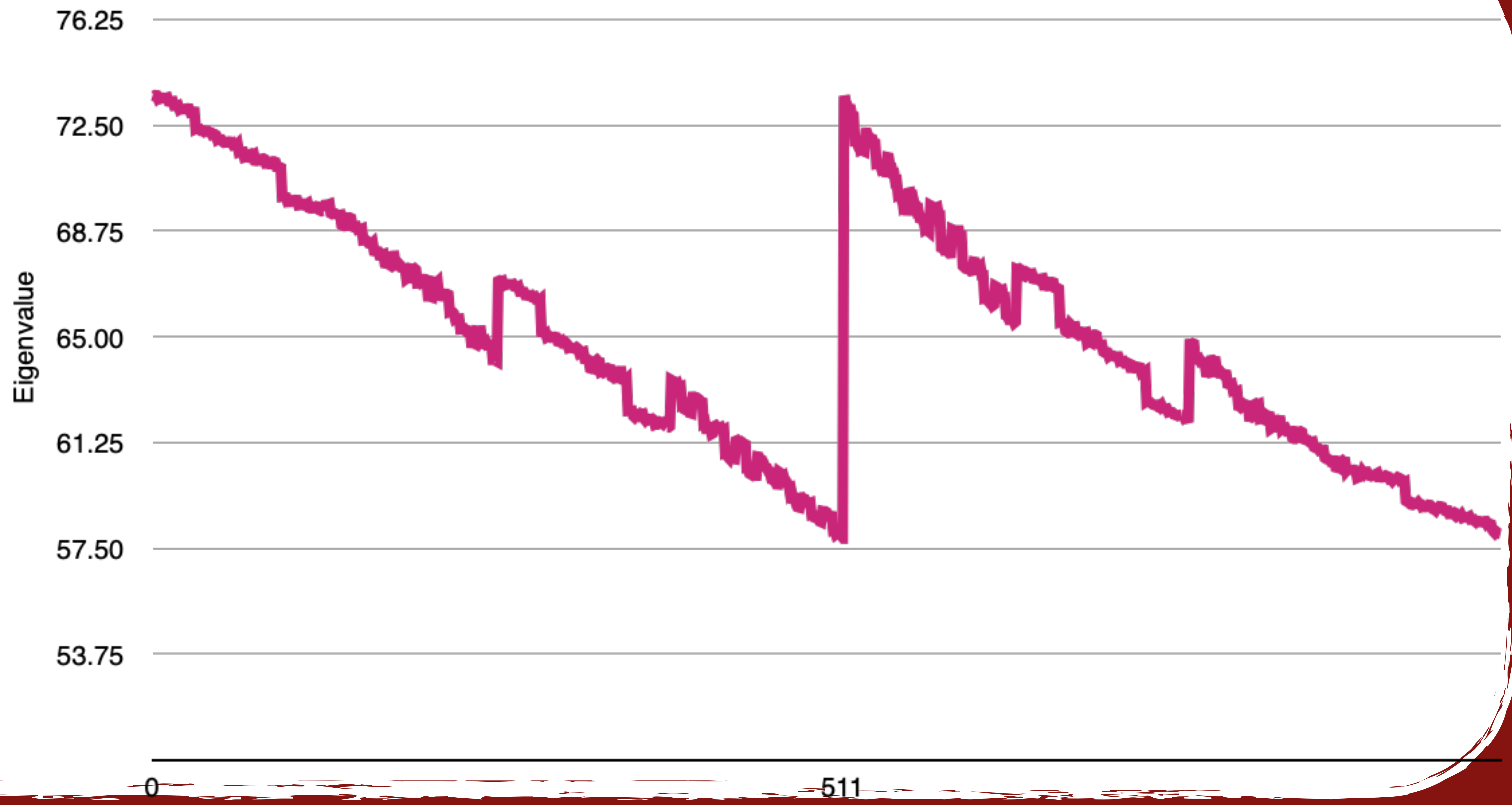
Advantage: Memory + CPU efficient (Billions of signatures can be processed)

Drawback: Does not work alone, eigenspaces are of dimension 2

Attack

Eigenvalue attack

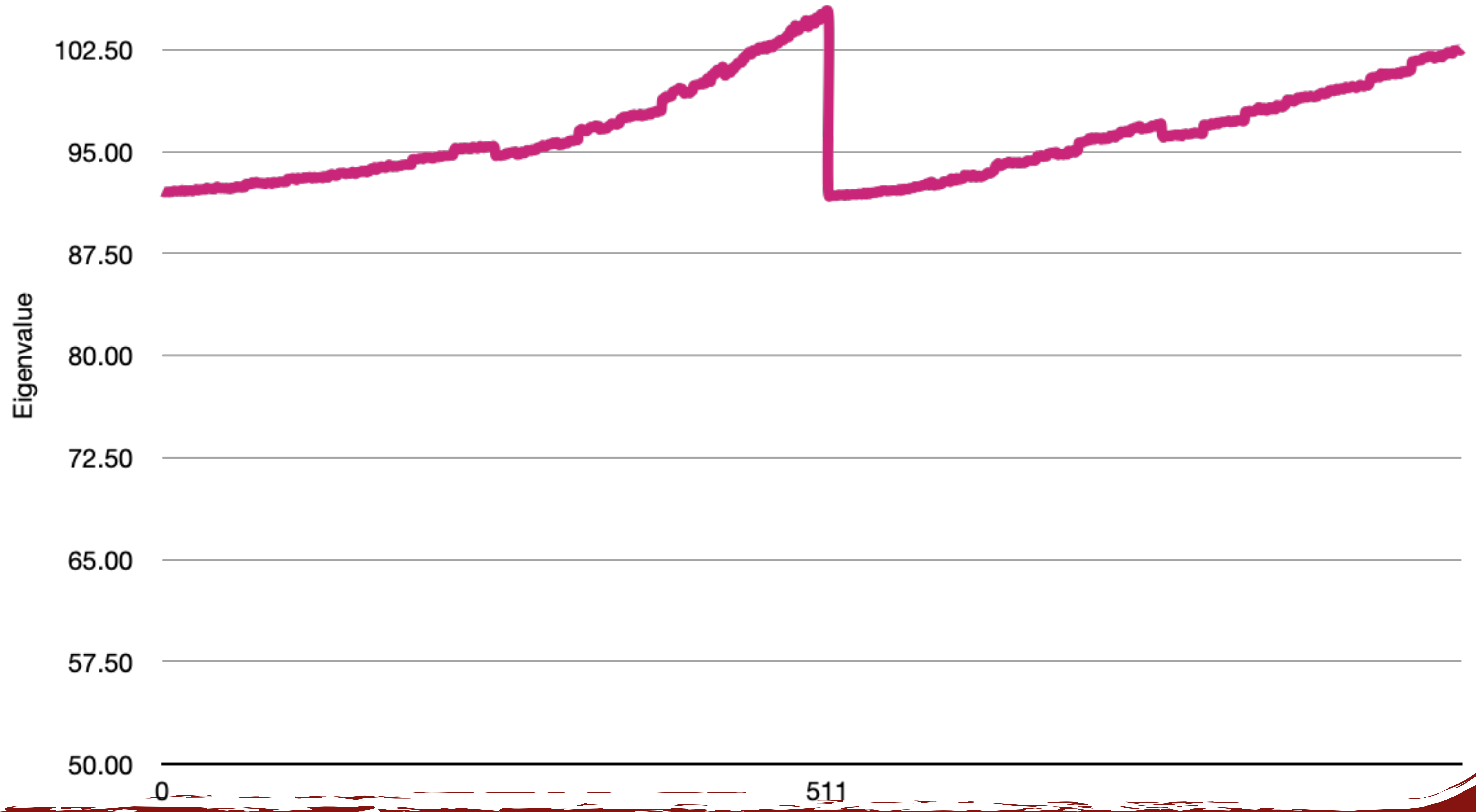
Distribution of **eigenvalues** of the **GSO vectors** for **8 bitflips**



Attack

Eigenvalue attack

Distribution of **eigenvalues** of the **GSO vectors** for **1 bitflip**



Attack

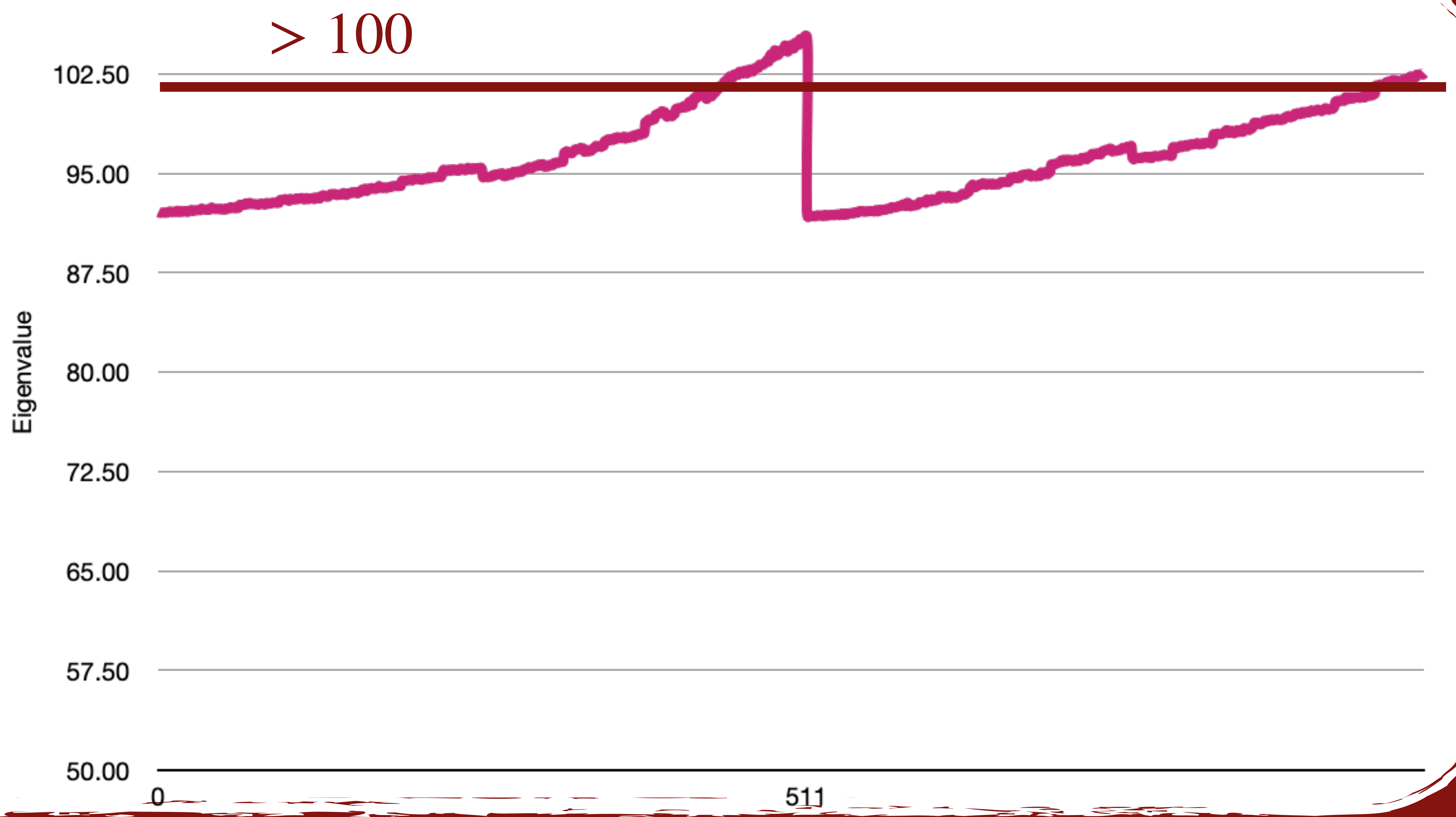
Summary: Shortcomings of NR06

- Bad results for full dimension (1024)
- **Observation**: Relevant eigenvectors live in small Subspace
- **Idea**: Perform search on Subspace
- **Problem**: How to find this subspace?

Attack

Eigenvalue attack

Distribution of **eigenvalues** of the **GSO vectors** for **1 bitflip**



Attack

Finding a good Subspace of dimension k

<u>Real Covariance</u>

$\Sigma, V = \text{Eigenspace}(\lambda_1, \ldots, \lambda_k)$

<u>Approximation</u>

$\hat{\Sigma}, \hat{V} = \text{Eigenspace}(\hat{\lambda}_1, \ldots, \hat{\lambda}_k)$

If $V = \hat{V}$, **project** signatures on $\hat{V}$ and perform NR06

Attack

Finding a good Subspace of dimension k

<u>Real Covariance</u>

$\Sigma, V = \text{Eigenspace}(\lambda_1, \ldots, \lambda_k)$

<u>Approximation</u>

$\hat{\Sigma}, \hat{V} = \text{Eigenspace}(\hat{\lambda}_1, \ldots, \hat{\lambda}_k)$

If $V = \hat{V}$, **project** signatures on $\hat{V}$ and perform NR06

But $V \neq \hat{V}$ in practice...

Attack

Finding a good Subspace of dimension k

Real Covariance

$\Sigma, V = \text{Eigenspace}(\lambda_1, \ldots, \lambda_k)$

Approximation

$\hat{\Sigma}, \hat{V} = \text{Eigenspace}(\hat{\lambda}_1, \ldots, \hat{\lambda}_k)$

If $V = \hat{V}$, **project** signatures on $\hat{V}$ and perform NR06

How "**close**" does it have to be?

Attack

(Variant of) Davis-Kahan theorem

$$\|\sin\Theta(\hat{V}, V)\|_{\mathrm{F}} \leq \frac{2\min(d^{1/2}\|\hat{\Sigma} - \Sigma\|_{\mathrm{op}}, \|\hat{\Sigma} - \Sigma\|_{\mathrm{F}})}{\min(\lambda_{r-1} - \lambda_r, \lambda_s - \lambda_{s+1})}.$$

Attack

(Variant of) Davis-Kahan theorem

$$\|\sin\Theta(\hat{V}, V)\|_{\mathrm{F}} \leq \frac{2\min(d^{1/2}\|\hat{\Sigma} - \Sigma\|_{\mathrm{op}}, \|\hat{\Sigma} - \Sigma\|_{\mathrm{F}})}{\min(\lambda_{r-1} - \lambda_r, \lambda_s - \lambda_{s+1})}.$$

$V = \text{Eigenspace}(\lambda_1, \ldots, \lambda_k)$

$\hat{V} = \text{Eigenspace}(\hat{\lambda}_1, \ldots, \hat{\lambda}_k)$

Davis-Kahan: subspaces are $\dfrac{\|\Sigma - \hat{\Sigma}\|}{\lambda_k - \lambda_{k+1}}$-close

Attack

(Variant of) Davis-Kahan theorem

$$\|\sin\Theta(\hat{V}, V)\|_F \le \frac{2\min(d^{1/2}\|\hat{\Sigma}-\Sigma\|_{op}, \|\hat{\Sigma}-\Sigma\|_F)}{\min(\lambda_{r-1}-\lambda_r, \lambda_s-\lambda_{s+1})}.$$

$V = \text{Eigenspace}(\lambda_1, \ldots, \lambda_k)$

$\hat{V} = \text{Eigenspace}(\hat{\lambda}_1, \ldots, \hat{\lambda}_k)$

But we only need **one** GSO vector to be in $\hat{V}$...

Attack

(Variant of) Davis-Kahan theorem

$$\|\sin\Theta(\hat{V}, V)\|_{\mathrm{F}} \le \frac{2\min(d^{1/2}\|\hat{\Sigma}-\Sigma\|_{\mathrm{op}}, \|\hat{\Sigma}-\Sigma\|_{\mathrm{F}})}{\min(\lambda_{r-1}-\lambda_r, \lambda_s-\lambda_{s+1})}.$$

$V = \text{Eigenspace}(\lambda_1, \ldots, \lambda_k)$

$\hat{V} = \text{Eigenspace}(\hat{\lambda}_1, \ldots, \hat{\lambda}_k)$

Our result: eigenvector v_1 is $\frac{\|\Sigma - \hat{\Sigma}\|}{\lambda_1 - \lambda_{k+1}}$-close to $\hat{V}$

Attack

Summary

2-step attack:

Compute a good approximation $\hat{\Sigma}$

Project signatures on $\hat{V}$ to perform NR

Results

Eigenvalue attack

How many bitflips to work?

	Nguyen-Regev Attack	**This work**
Full flip (Empty table)	✓	✓
8 bitflips	✗	✓
1 bitflip	✗	✓

Results

Eigenvalue attack

Efficiency?

	Nguyen-Regev Attack	**This work**
Full flip (Empty table)	**2M**	✓
8 bitflips	✗	**20M + 2M ($k < 16$)**
1 bitflip	✗	**300M + 20M ($k < 64$)**

Results

Countermeasures

- Bitflips **reduce** signature sizes
- Bitflips are **permanent** in RAM

Results

Countermeasures

- Bitflips **reduce** signature sizes
- Bitflips are **permanent** in RAM

Lower Bound Rejection

Integrity Check

Conclusion

What next?

- Extend the attack to other RCDT-based schemes (Hawk)
- Find other ways to finish the attack

Thanks for watching!