Mini-symposium on Sensor Network Localization and Dynamical Distance Geometry

FIELDS

UNIVERSITÉ DE RENNES 1

UFSC

Optically solving the **Distance Geometry Problem***

NARA Rubiano da Silva

Department of Physics, Federal University of Santa Catarina (Brazil)

Simon B. Hengeveld, Douglas S. Gonçalves, Paulo Henrique Souto Ribeiro, and Antonio Mucherino

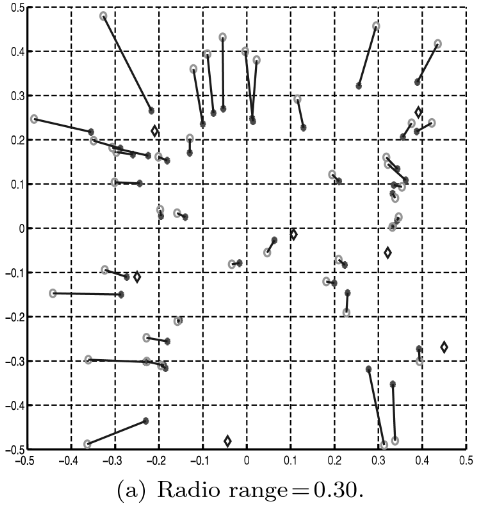
Department of Mathematics, Federal University of Santa Catarina (Brazil)

IRISA, University of Rennes (France)

*arXiv:2105.12118

Motivation

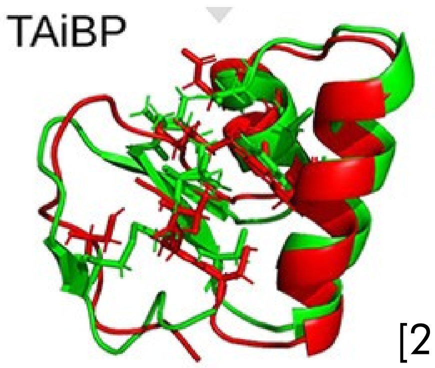
Sensor network localization



(a) Radio range = 0.30.

[1]

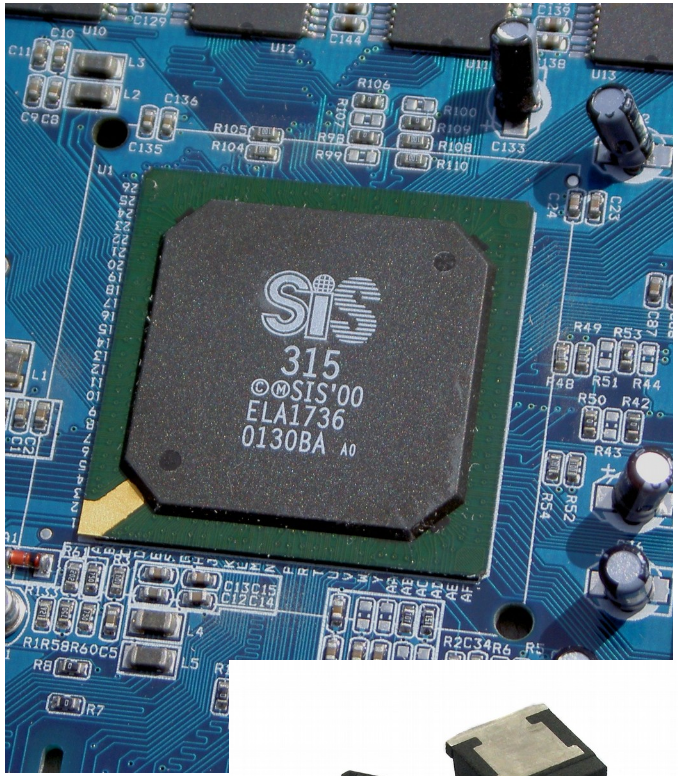
Molecular conformation

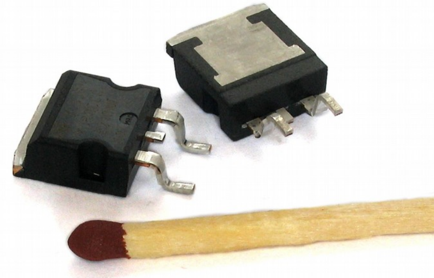


[2]

Distance Geometry Problem

New computing platforms





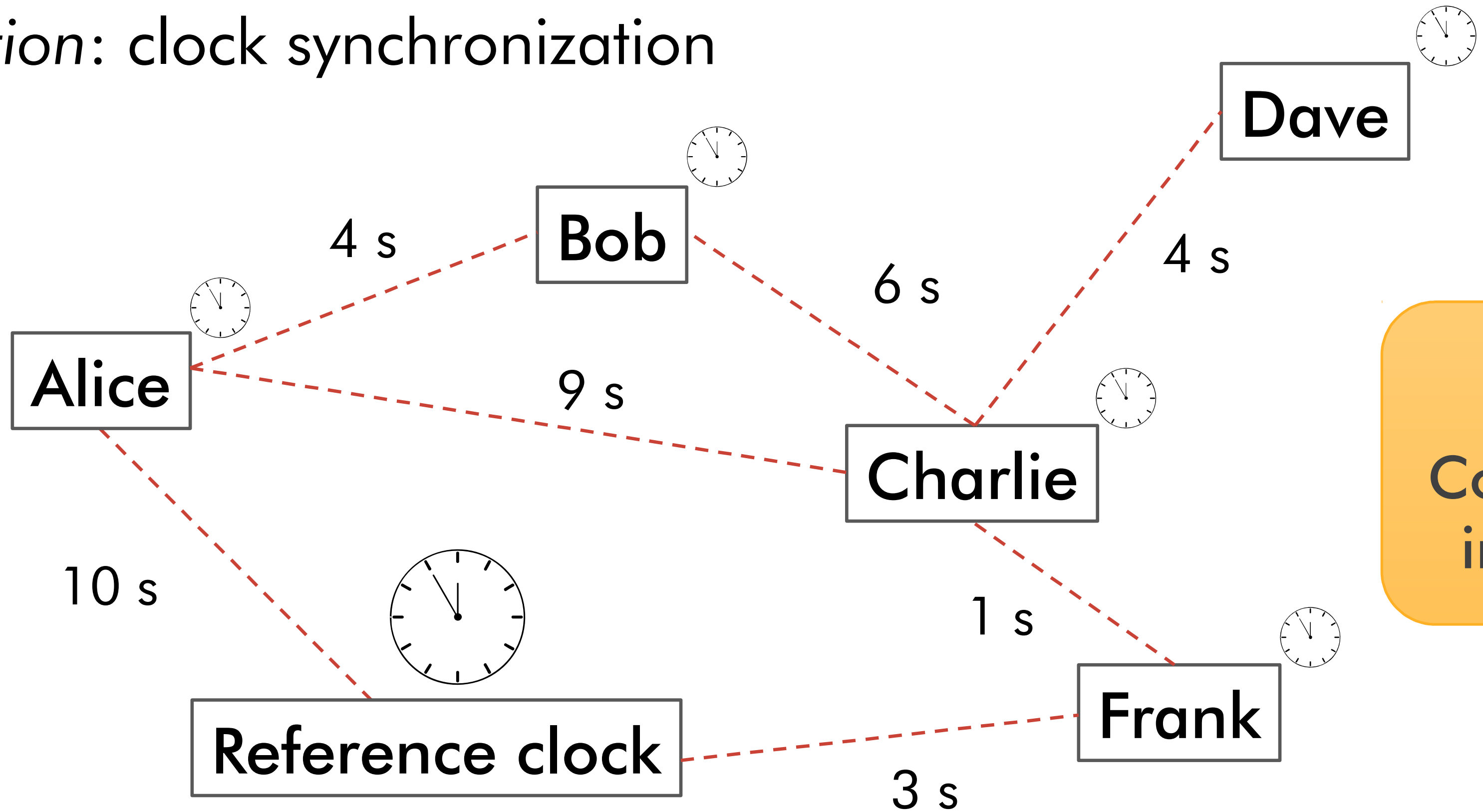
[1] Biswas *et al.*, ACM Trans. Sens. Netw. **2** (2006)

[2] Malliavin *et al.*, J. Chem. Inf. Model. **59** (2019)

Which problem?

Distance Geometry Problem in dimension 1

Application: clock synchronization



NP-hard!

Complexity of algorithms increases exponentially

A. Singer, *Appl. Comput. Harmon. A* **30** (2011)

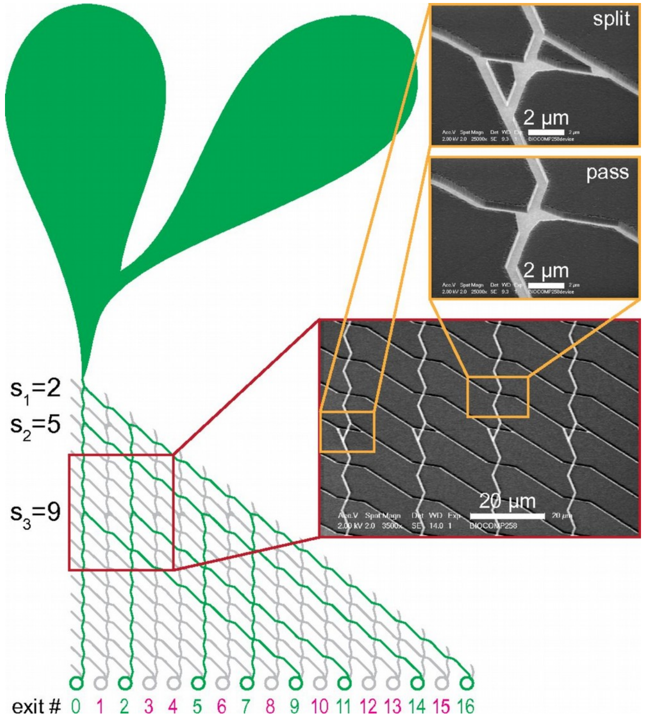
Which platform?

Solving Subset-Sum Problem in different platforms

Parallel computing

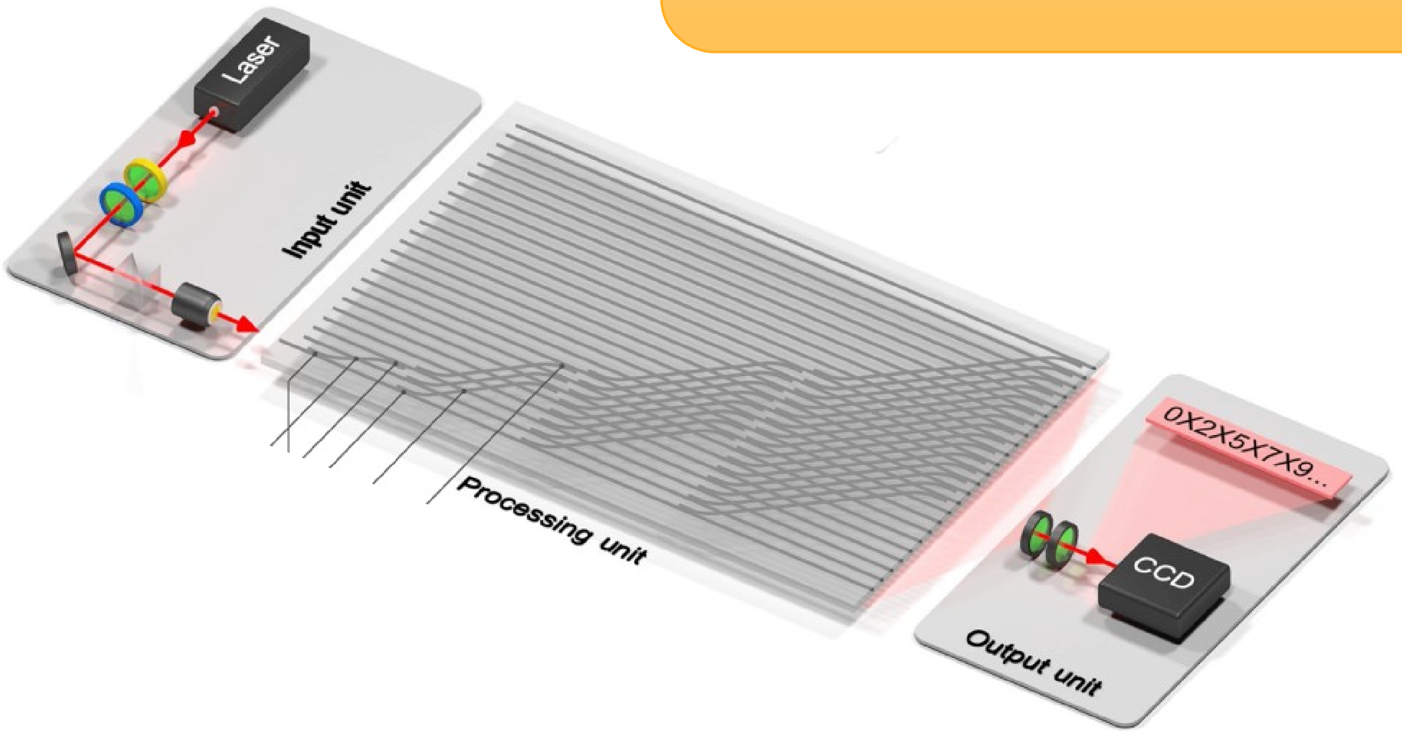
Reduce algorithm complexity

Molecular network



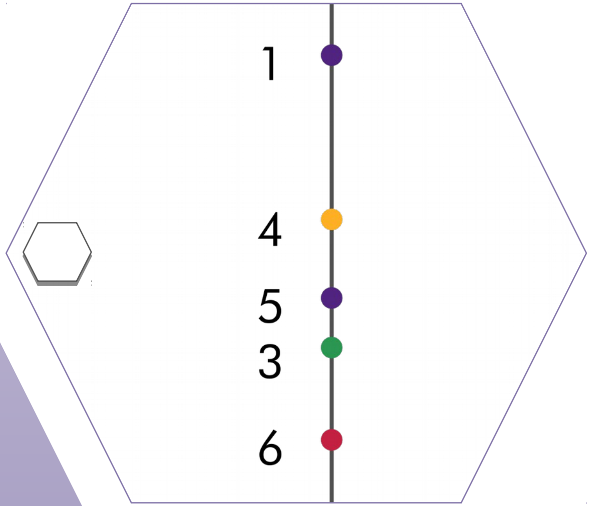
Nicolau Jr. *et al.*, *PNAS* **113** (2016)

Optical network



Xu *et al.*, *Science Adv.* **6** (2020)

Outline

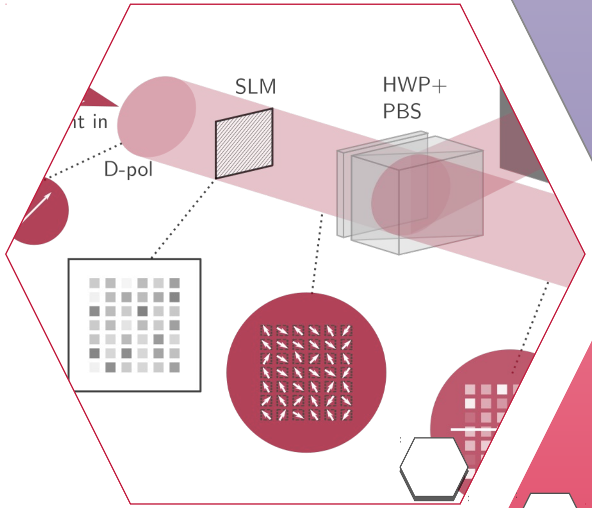


Paradoxical Distance Geometry Problem

Matrix-by-vector reformulation

$$= \begin{pmatrix} -1 & -1 & -1 \\ 1 & -1 & -1 \\ -1 & 1 & -1 \\ 1 & 1 & -1 \\ -1 & -1 & 1 \\ 1 & -1 & 1 \\ -1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

Optical scheme



Paradoxical DGP

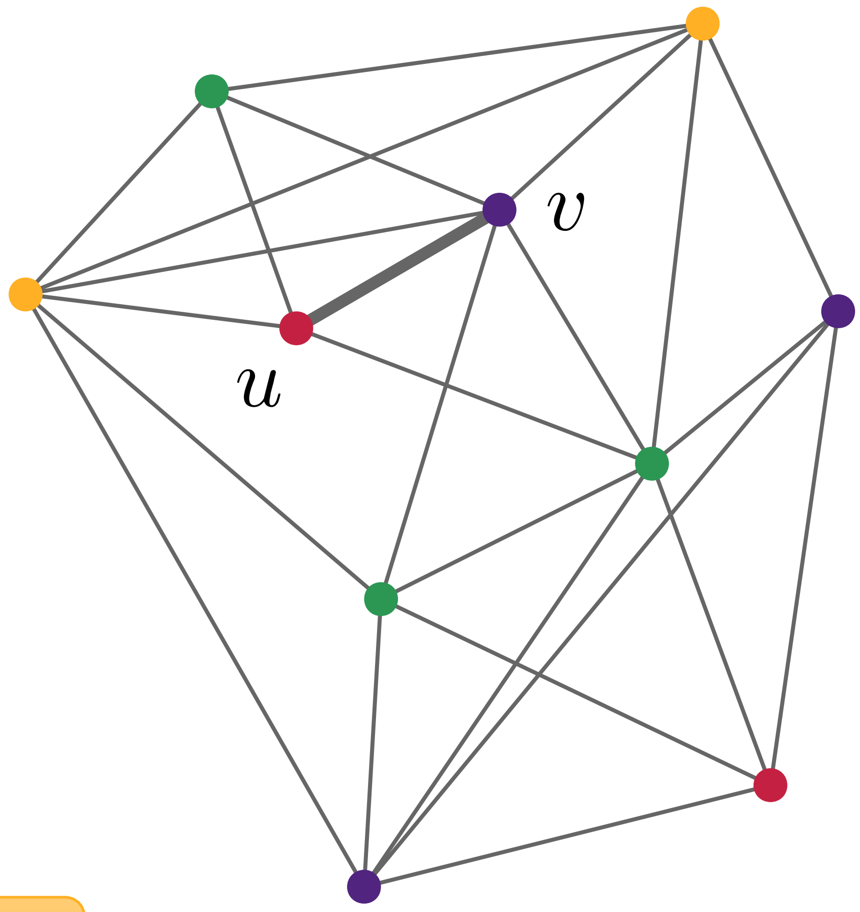
Distance Geometry Problem

Graph $G = (V, E, d)$

Vertices V

Edges E

Distances d *



* Weight $d(u, v) \in \mathbb{R}_+$ when $\{u, v\} \in E$

NP-hard!

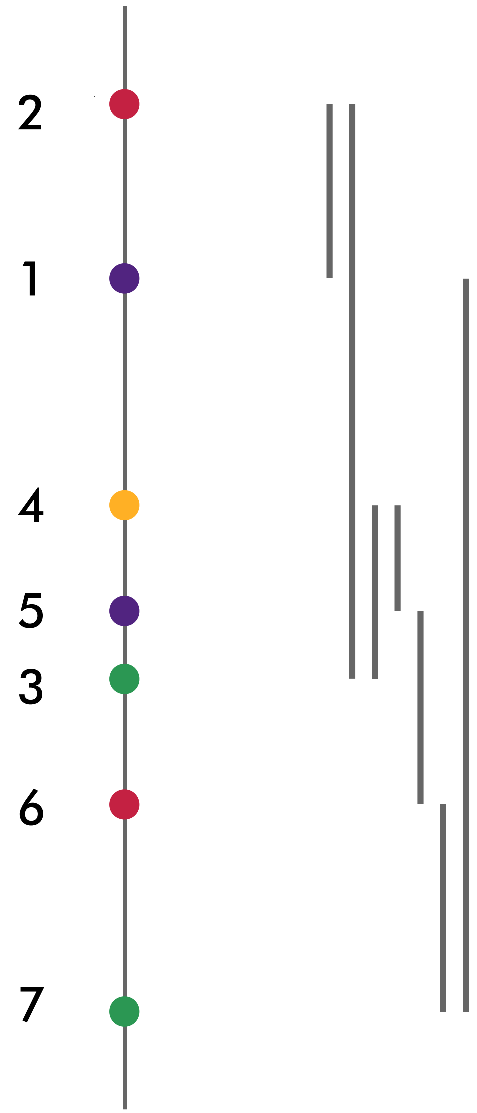
Paradoxical DGP_1

DGP_1 : distance geometry problem in dimension 1

Definition of paradoxical DGP_1

i. Vertex order exists: for every vertex (different from the first), the edge k $\{k-1, k\}$ belongs to E;

ii. No other edge is in E except $\{1, n\}$ (distance from first to last vertex in the order, $n = |V|$).



Paradoxical DGP_1

DGP_1 : distance geometry problem in dimension 1

Definition of paradoxical DGP_1

i. Vertex order exists: for every vertex (different from the first), the edge k $\{k-1, k\}$ belongs to E;

ii. No other edge is in E except $\{1, n\}$ (distance from first to last vertex in the order, $n = |V|$).

- Presence of $d(1, n)$ → **only two solutions** are possible;
- $d(1, n)$ involves the very first and the very last vertex → information can be exploited **only in the last moment.**

BuildUp algorithms

Construct possible DGP solutions:

i. One vertex position at a time;

ii. Predefined vertex order.

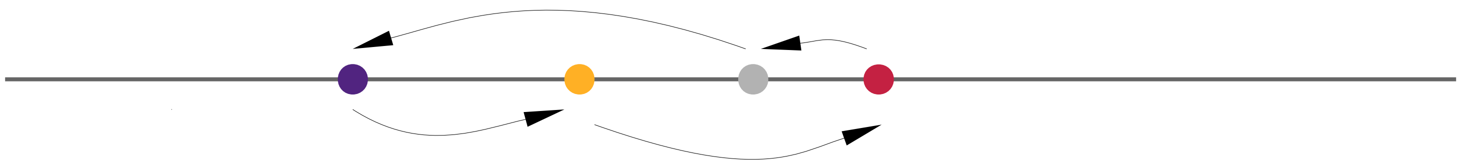
→ can be used to generate a search tree (multiple realizations)

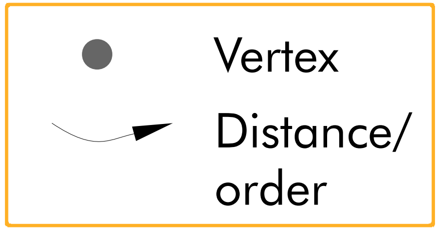
For paradoxical DGP_1

- **# of potential realizations keeps exponentially growing** until reaching the last vertex;
- at most two realizations are feasible (correspond to all distances)

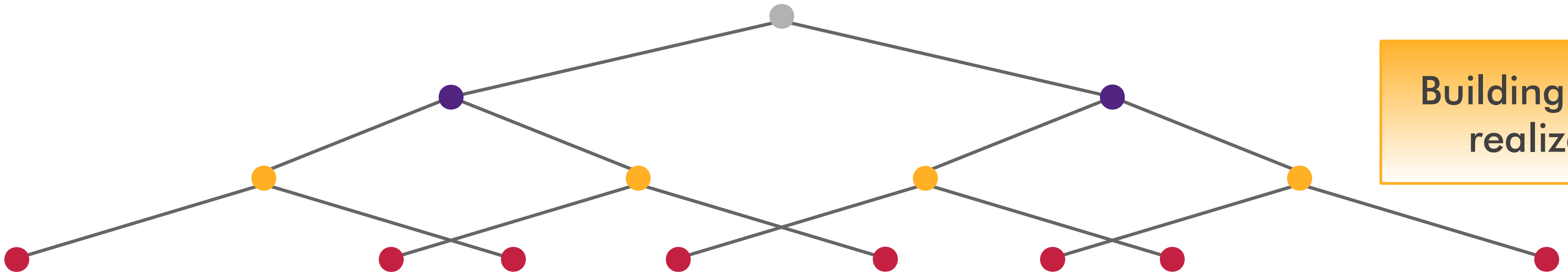
Branch-and-Prune I

Expected solution



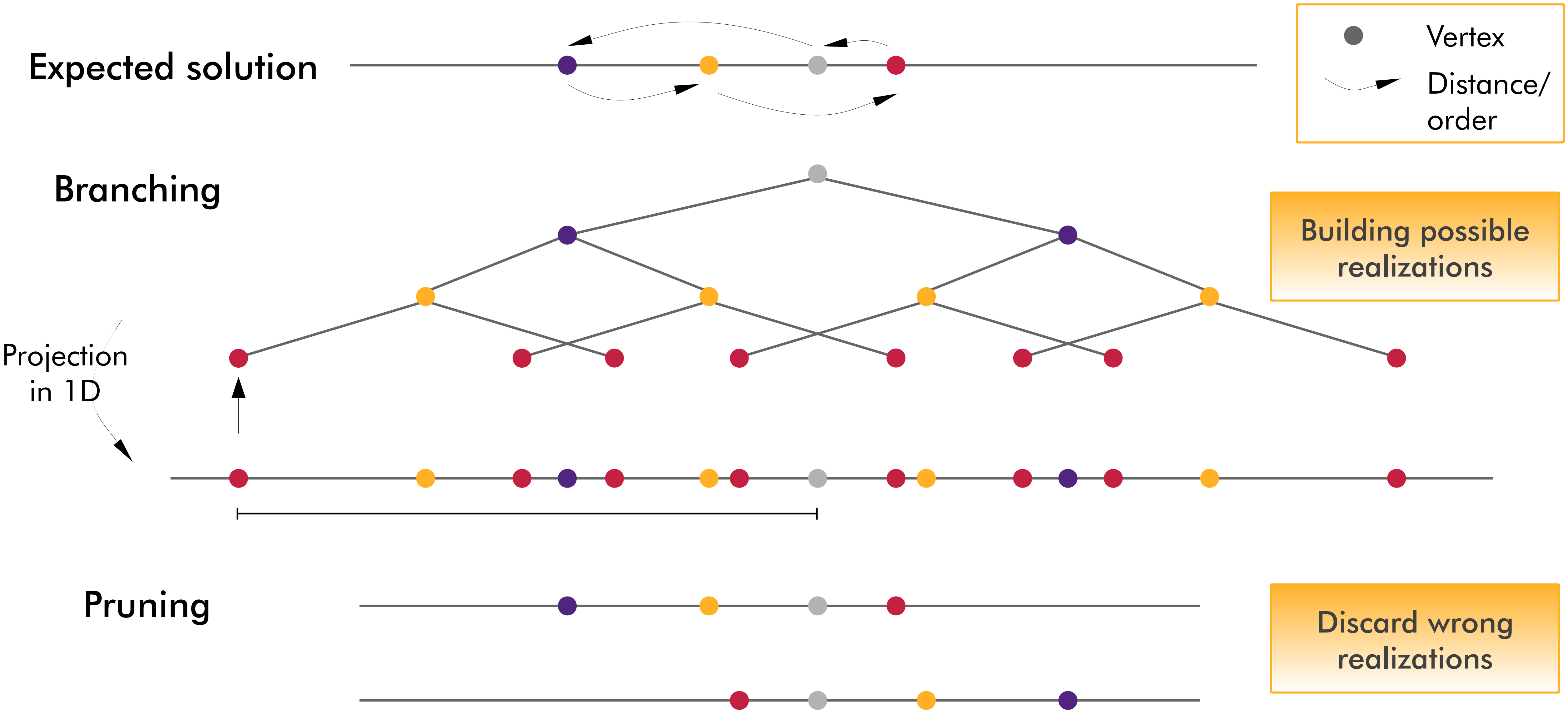


Branching



Building possible realizations

Branch-and-Prune I



Branch-and-Prune II

For a paradoxical DGP$_1$ instance:

i. Initialize position of vertex 1: $x_1 = 0$

ii. Possible positions for all subsequent vertices:

$$x_k = x_{k-1} + s_k d_{k-1,k}, \quad \forall k = 2, \ldots, n,$$
$$d_{k-1,k} = d(k-1, d); \qquad s_k = -1 \ \text{ or } \ s_k = +1$$

iii. Branching: computing $2p$ positions for the vertex k (p is the number of available positions for the preceding vertex).

iv. Pruning: at the last layer n, $d_{1,n}$ is used to select only two solutions of the tree.

HERE: uniformize branching/pruning by including a "virtual" vertex $n{+}1$, for which $d_{n,n+1} = d_{1,n}$

→ feasibility of a solution: $x_1 = x_{n+1}$

Matrix-by-vector reformulation

Basic idea

$$x_k = x_{k-1} + s_k d_{k-1,k}, \quad \forall k = 2, \ldots, n,$$
$$s_k = -1 \;\; \text{or} \quad s_k = +1, \quad \text{and} \quad x_1 = 0$$

- A solution is represented by a boolean n-vector (s_k): $(-1, +1, +1, \ldots, -1, -1)$
- … when multiplied by a vector containing the distances: $\mathbf{y} = (d_{12}, d_{23}, \ldots, d_{n-1,n}, d_{1n})^T$

  → position of vertex $n+1$ (check feasibility)

$$x_{k+1} = (-1, +1, \ldots, -1, -1) \begin{bmatrix} d_{12} \\ d_{23} \\ \vdots \\ d_{n-1,n} \\ d_{1n} \end{bmatrix} = -d_{12} + d_{23} + \ldots - d_{n-1,n} - d_{1n}$$

Binary matrix

- Take all possible realizations at once → binary matrix $\mathbf{M}$:
  - 2^n rows – potential realizations with fictive vertex
  - n columns – n vertices

$$x_1 = 0$$

- Only one vertex in G: $\mathbf{M} = \begin{pmatrix} -1 \\ 1 \end{pmatrix}, \quad \mathbf{y} = (d_{11}) \quad \Rightarrow \mathbf{r} = \begin{pmatrix} -d_{11} \\ d_{11} \end{pmatrix}.$

- Two vertices in G: $\mathbf{M} = \begin{pmatrix} -1 & -1 \\ 1 & -1 \\ -1 & 1 \\ 1 & 1 \end{pmatrix}, \quad \mathbf{y} = \begin{pmatrix} d_{12} \\ d_{12} \end{pmatrix} \quad \Rightarrow \mathbf{r} = \begin{pmatrix} -d_{12} - d_{12} \\ d_{12} - d_{12} \\ -d_{12} + d_{12} \\ +d_{12} + d_{12} \end{pmatrix}.$

Pattern length: 2^j

Binary matrix

- Take all possible realizations at once → binary matrix $\mathbf{M}$:
  - 2^n rows – potential realizations with fictive vertex
  - n columns – n vertices

$x_1 = 0$

- Only one vertex in G: $\mathbf{M} = \begin{pmatrix} -1 \\ 1 \end{pmatrix}, \quad \mathbf{y} = (d_{11}) \quad \Rightarrow \mathbf{r} = \begin{pmatrix} -d_{11} \\ d_{11} \end{pmatrix}.$
- Three vertices:

$$\mathbf{M} = \begin{pmatrix} -1 & -1 & -1 \\ 1 & -1 & -1 \\ -1 & 1 & -1 \\ 1 & 1 & -1 \\ -1 & -1 & 1 \\ 1 & -1 & 1 \\ -1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}, \quad \mathbf{y} = \begin{pmatrix} d_{12} \\ d_{23} \\ d_{13} \end{pmatrix} \quad \Rightarrow \mathbf{r} = \begin{pmatrix} -d_{12} - d_{23} - d_{13} \\ d_{12} - d_{23} - d_{13} \\ -d_{12} + d_{23} - d_{13} \\ d_{12} + d_{23} - d_{13} \\ -d_{12} - d_{23} + d_{13} \\ d_{12} - d_{23} + d_{13} \\ -d_{12} + d_{23} + d_{13} \\ d_{12} + d_{23} + d_{13} \end{pmatrix}$$

Pattern length: 2^j

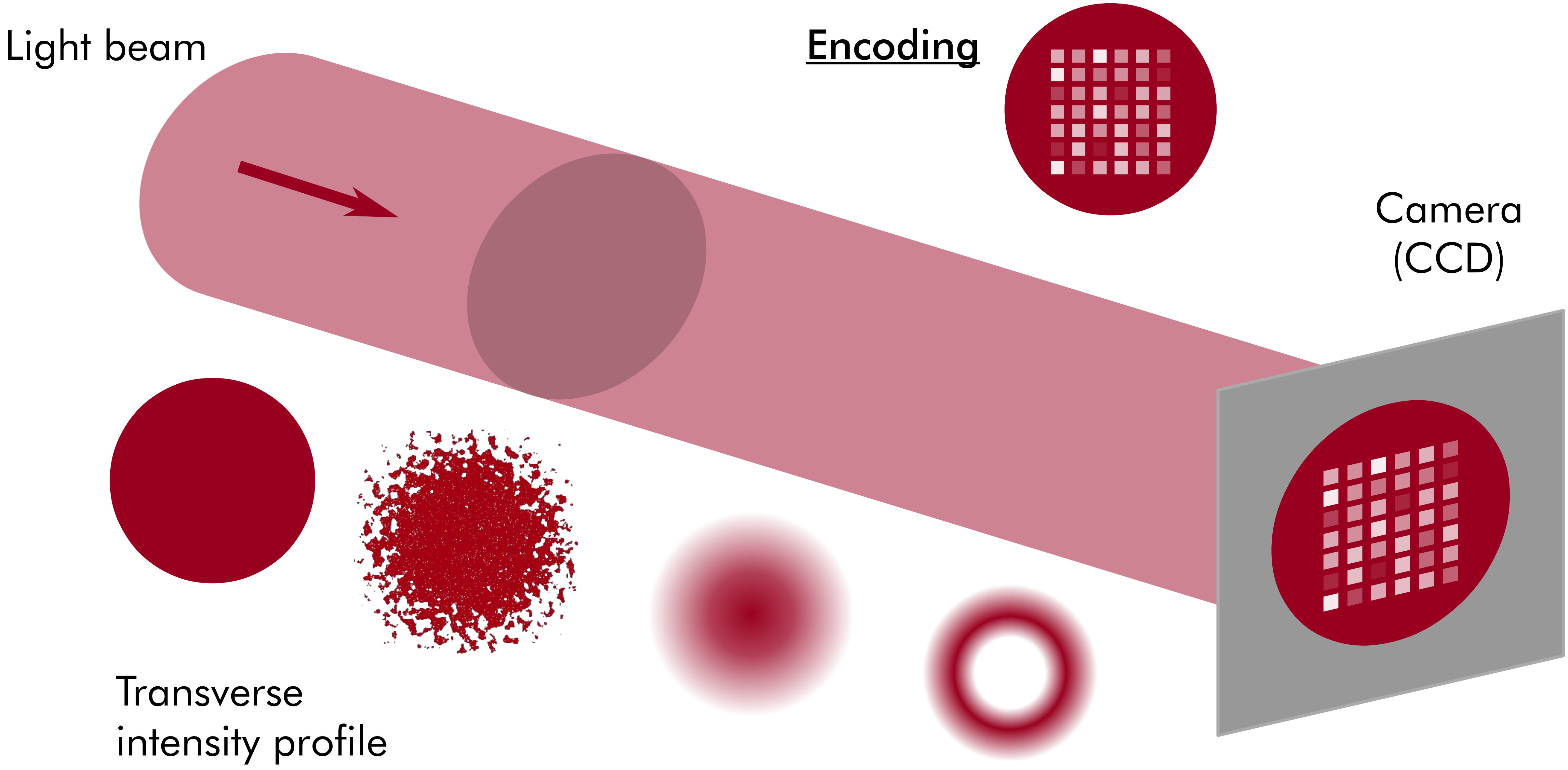
Matrix-by-vector multiplication reformulation

- Realizations vector: $\mathbf{r} = \mathbf{M} \times \mathbf{y}$
- Distance vector: $\mathbf{y} = \begin{bmatrix} d_{12} \\ d_{23} \\ \vdots \\ d_{n-1,n} \\ d_{1n} \end{bmatrix}$
- Binary matrix: $\mathbf{M}_{ij} = \begin{cases} -1 & \text{if } (i-1)2^{1-j} \bmod 2 = 0, \\ 1 & \text{otherwise.} \end{cases}$

Feasible solutions: null components of $\mathbf{r}$

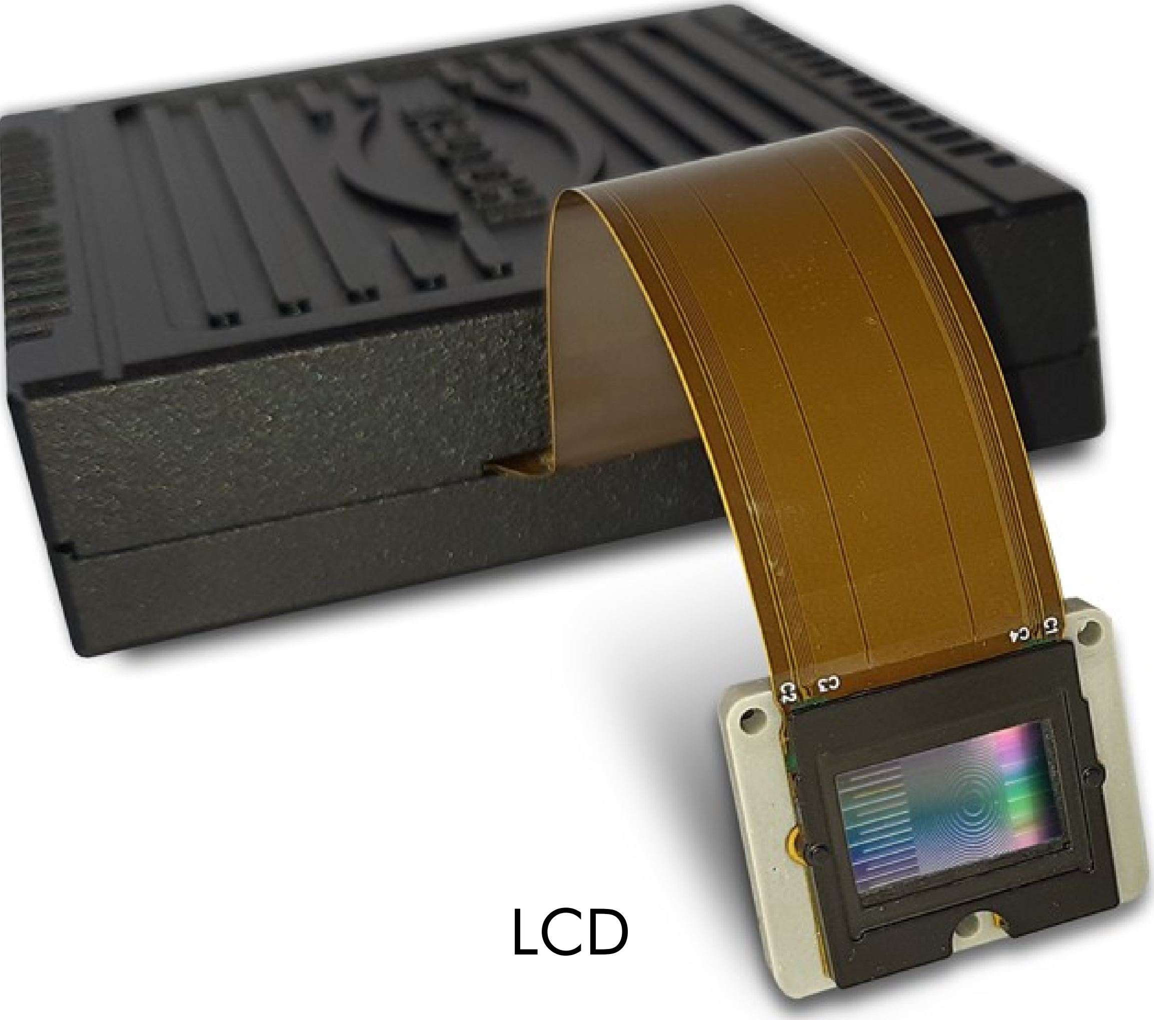
Optical scheme

Optical matrices?

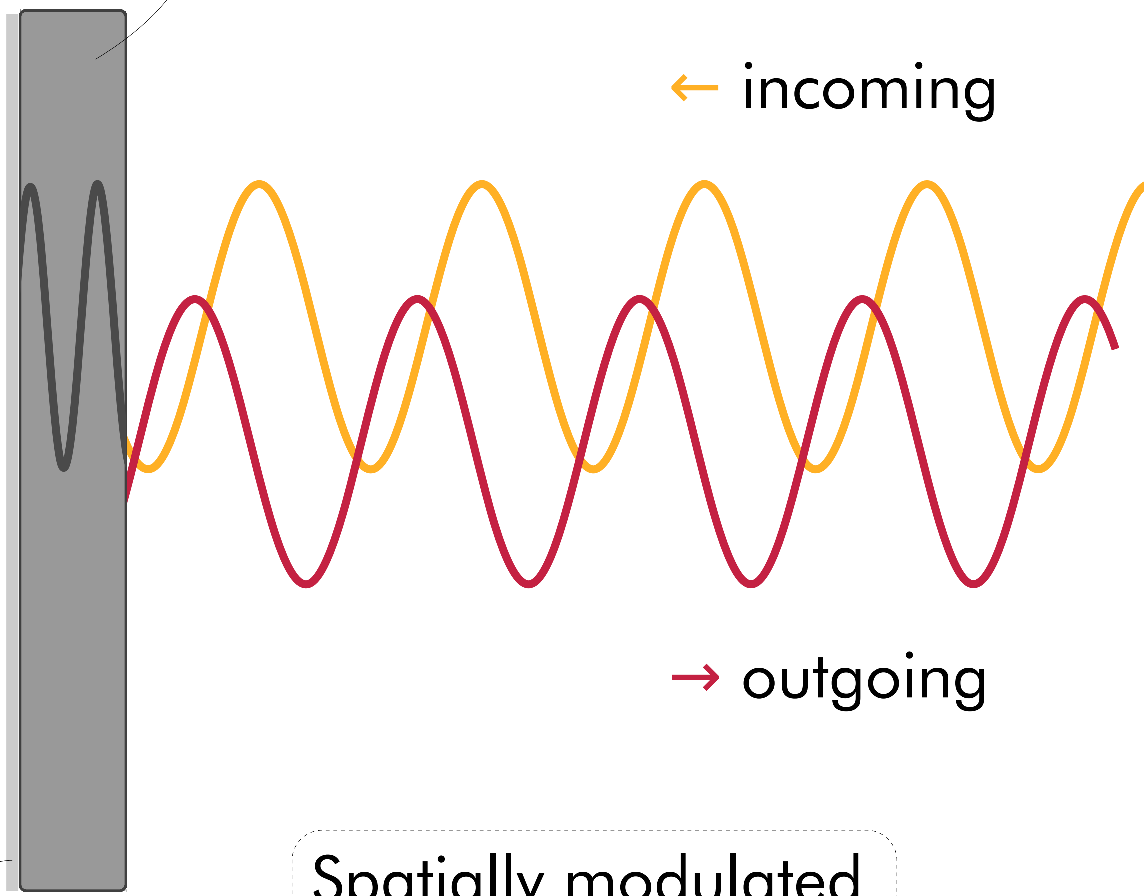


Encoding scheme I

SLM = spatial light modulator



LCD



Spatially modulated
DEPHASING

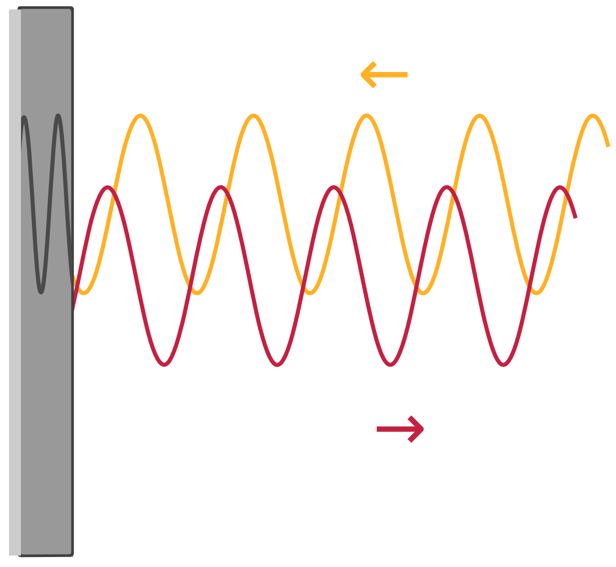
https://holoeye.com/gaea-4k-phase-only-spatial-light-modulator/

Encoding scheme II

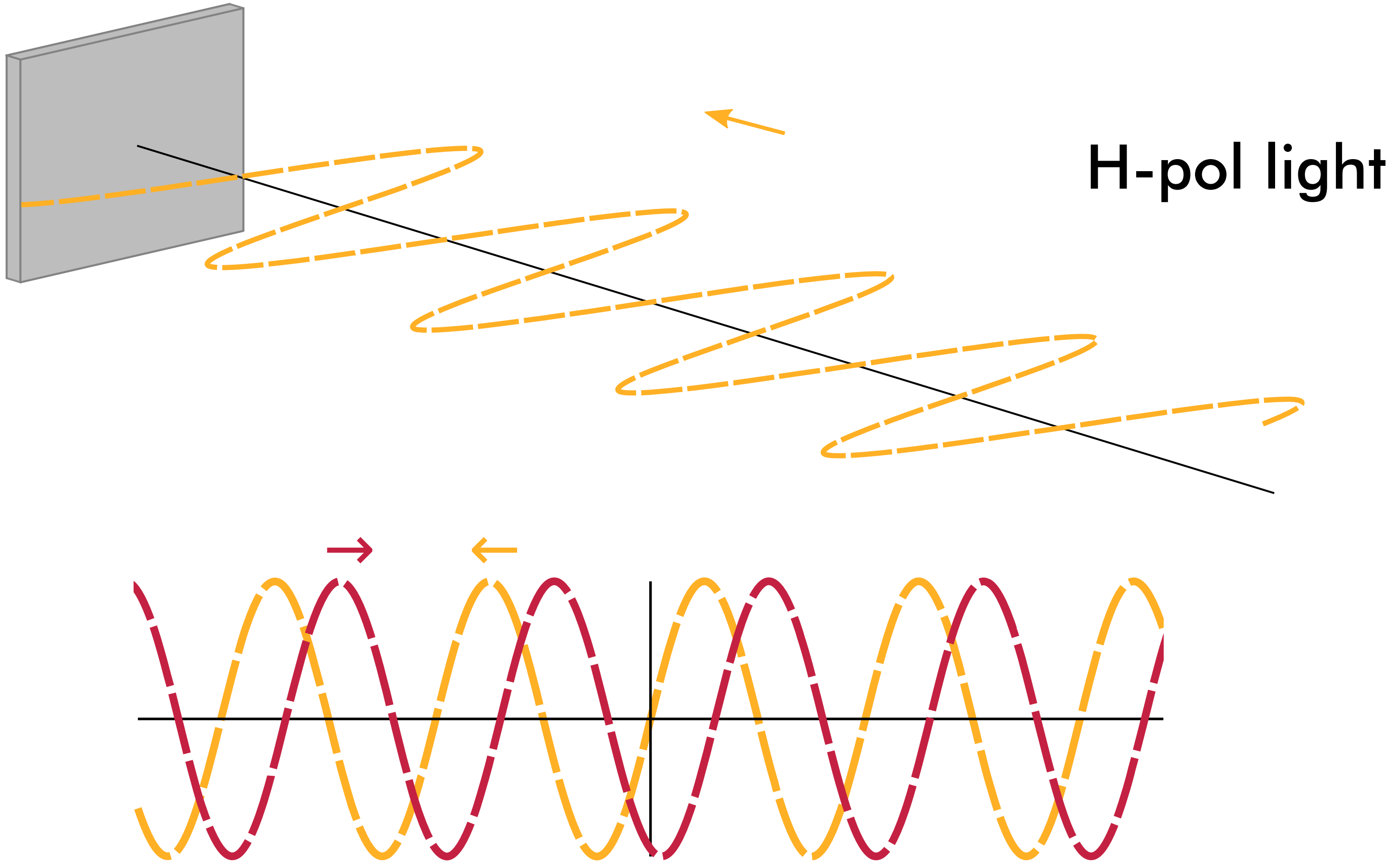
Spatially modulated
DEPHASING

?

Spatially modulated
INTENSITY

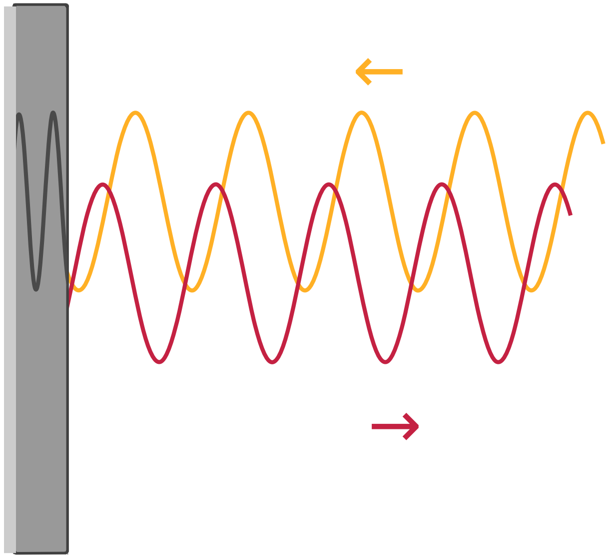


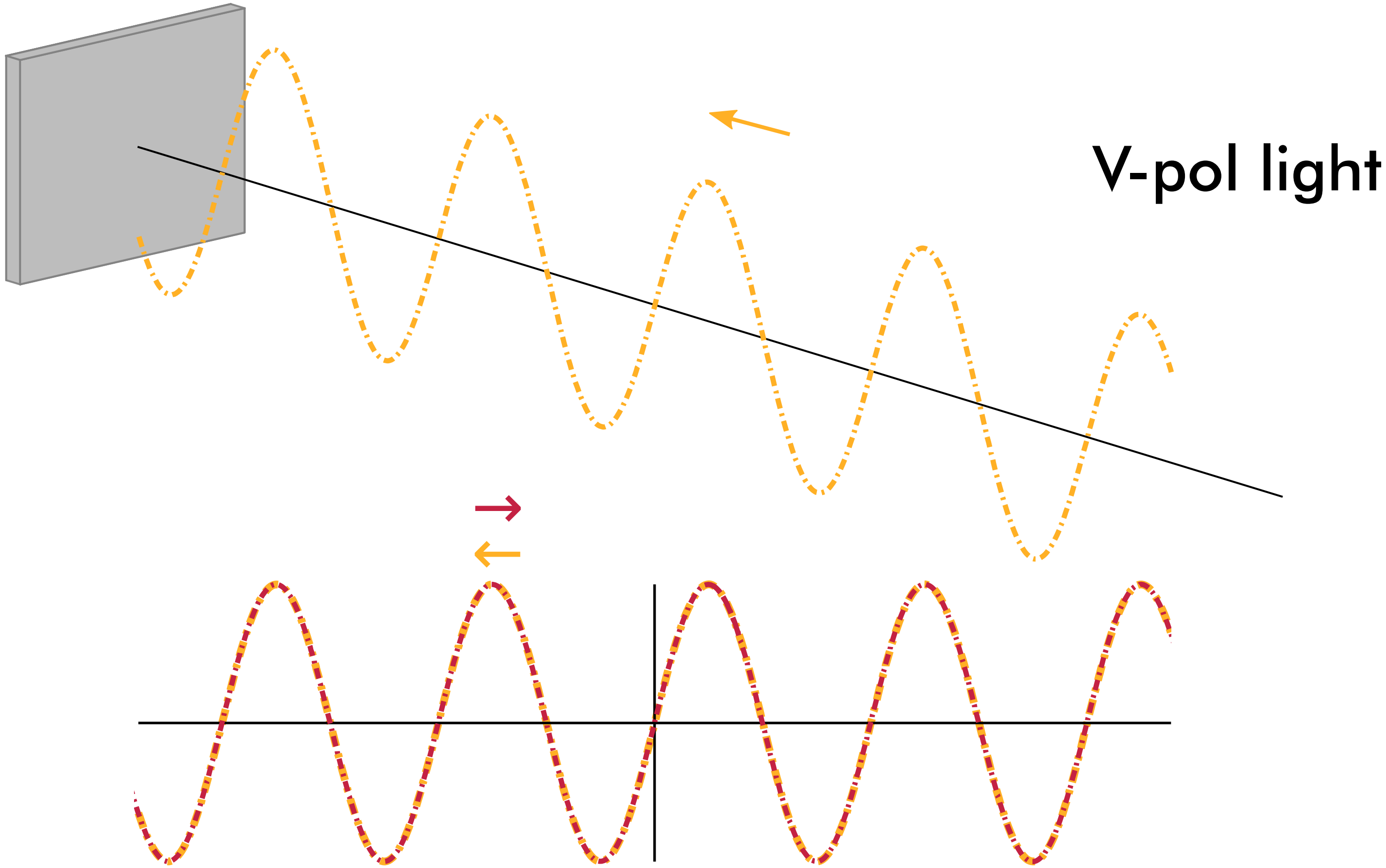
Polarization-dependent



Encoding scheme II

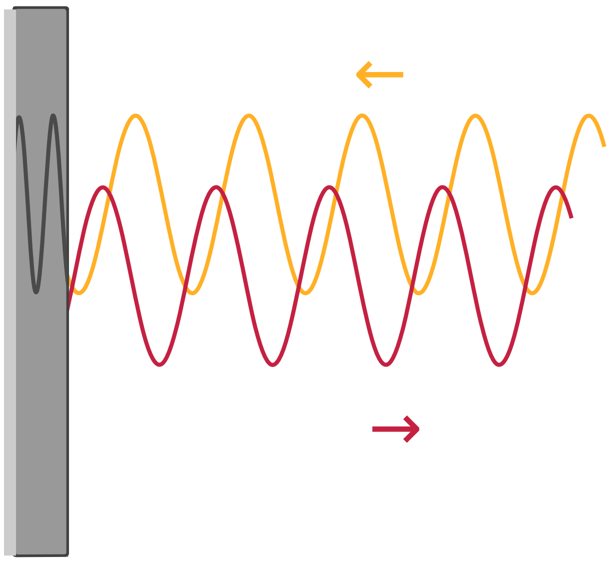
Spatially modulated **DEPHASING** → ? → Spatially modulated **INTENSITY**


Polarization-dependent

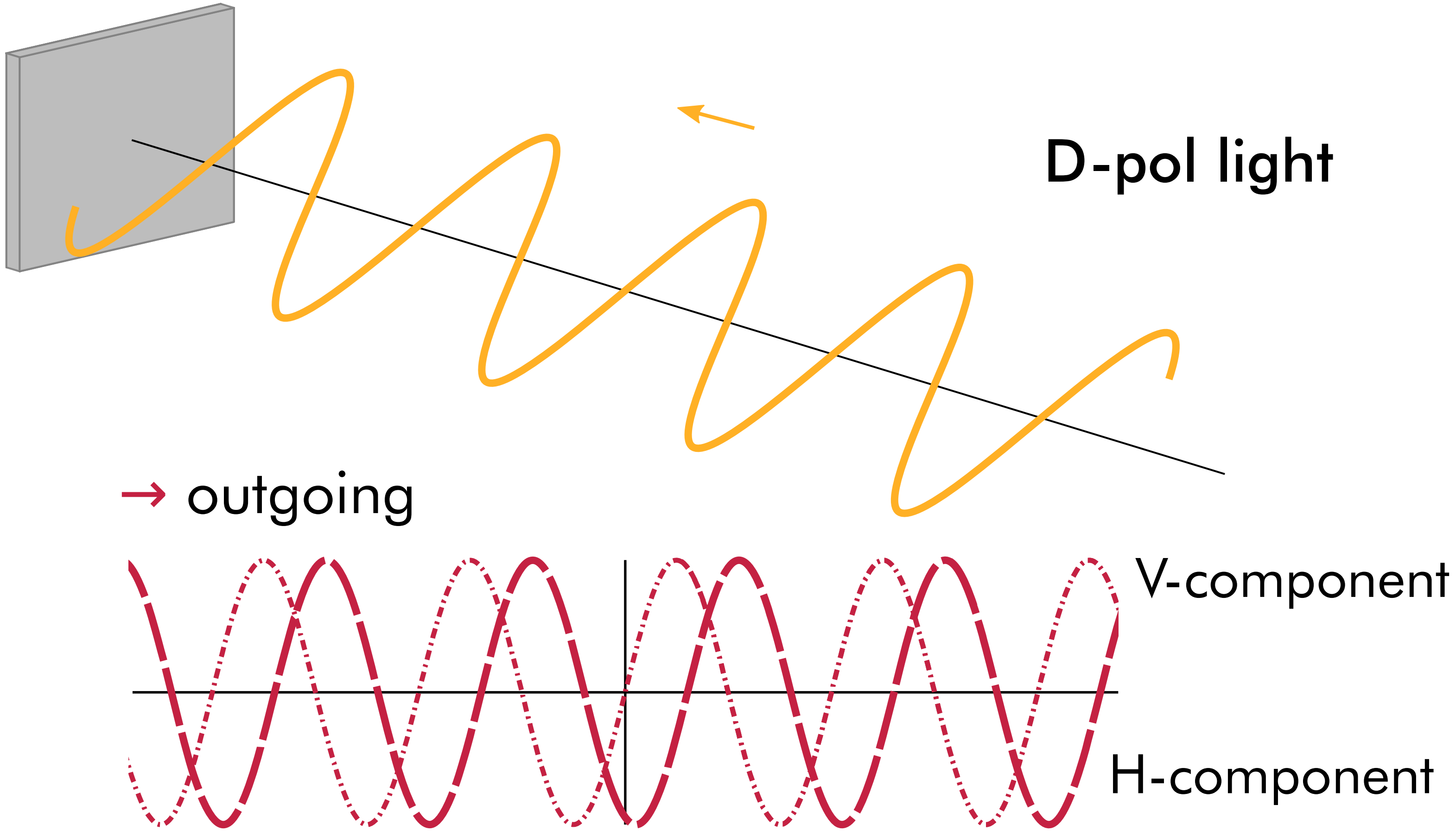


Encoding scheme II

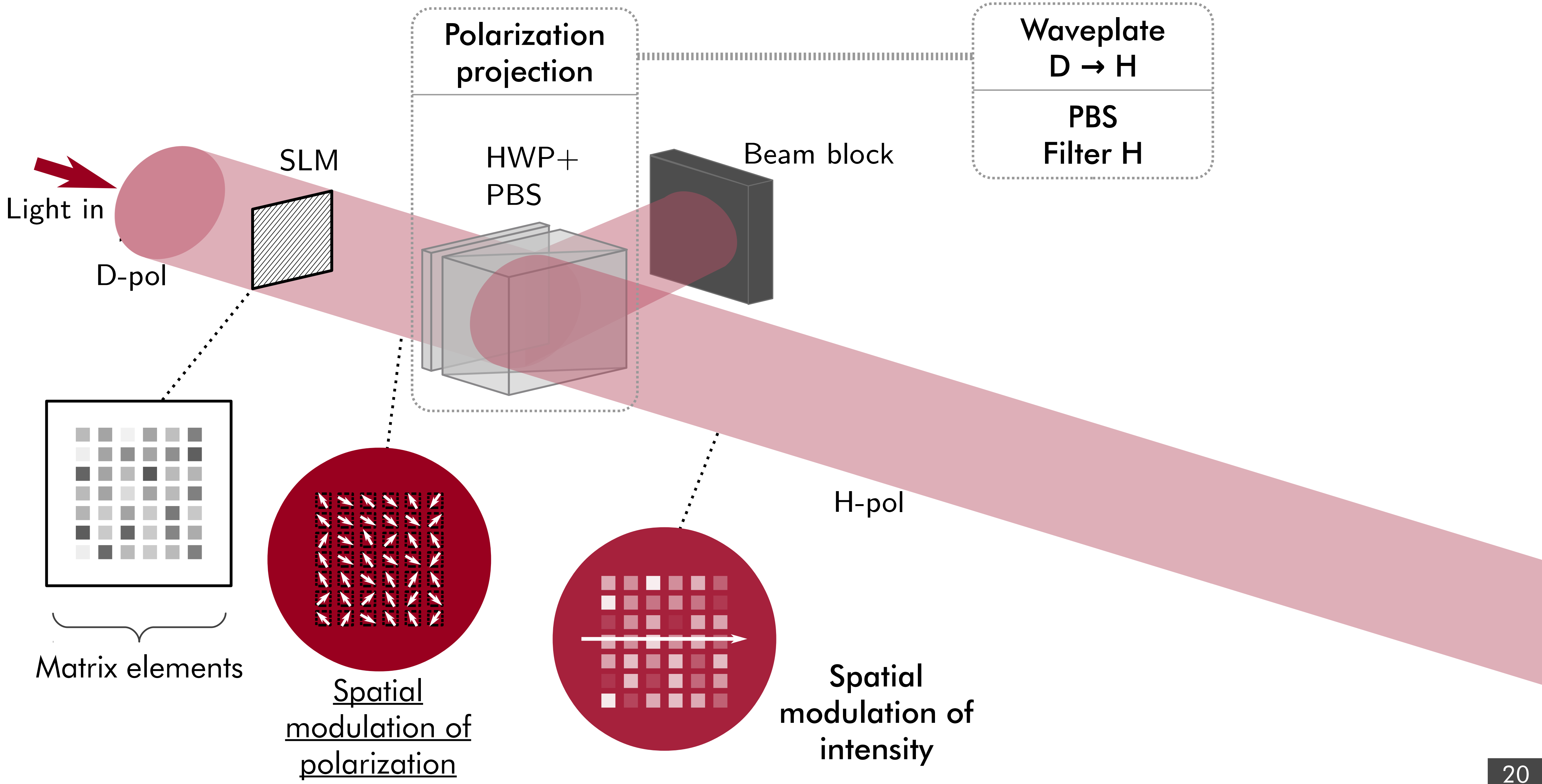
Spatially modulated **DEPHASING** → Use D-pol light (in) Filter D-pol (out) → Spatially modulated **INTENSITY**



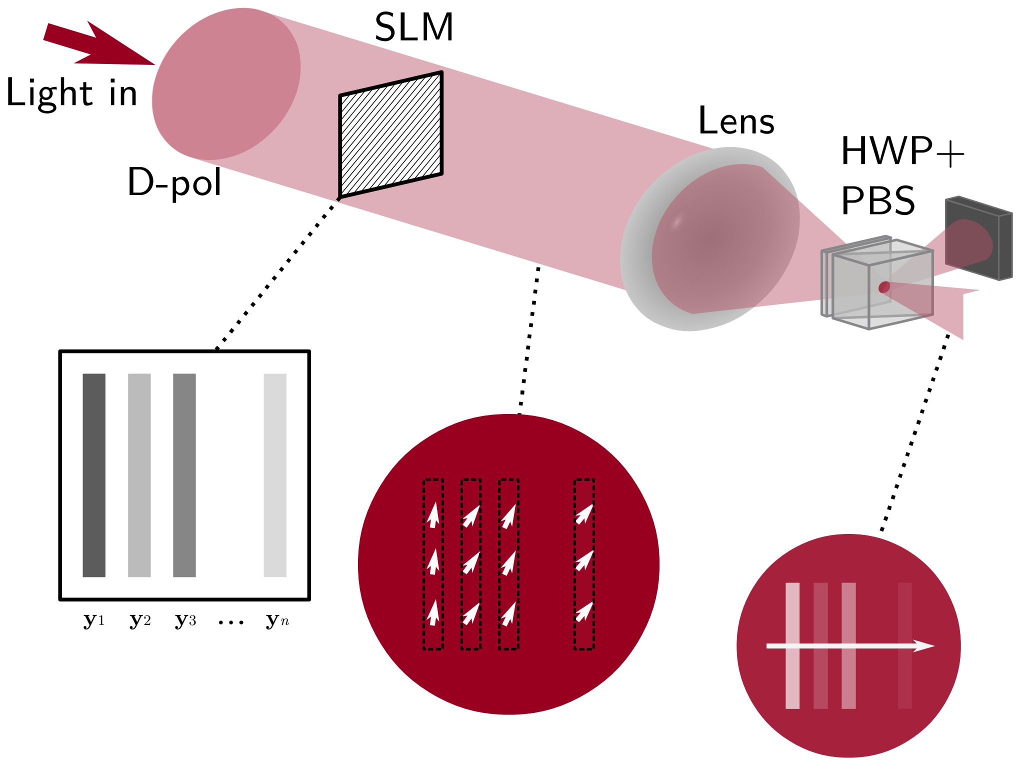
Polarization-dependent

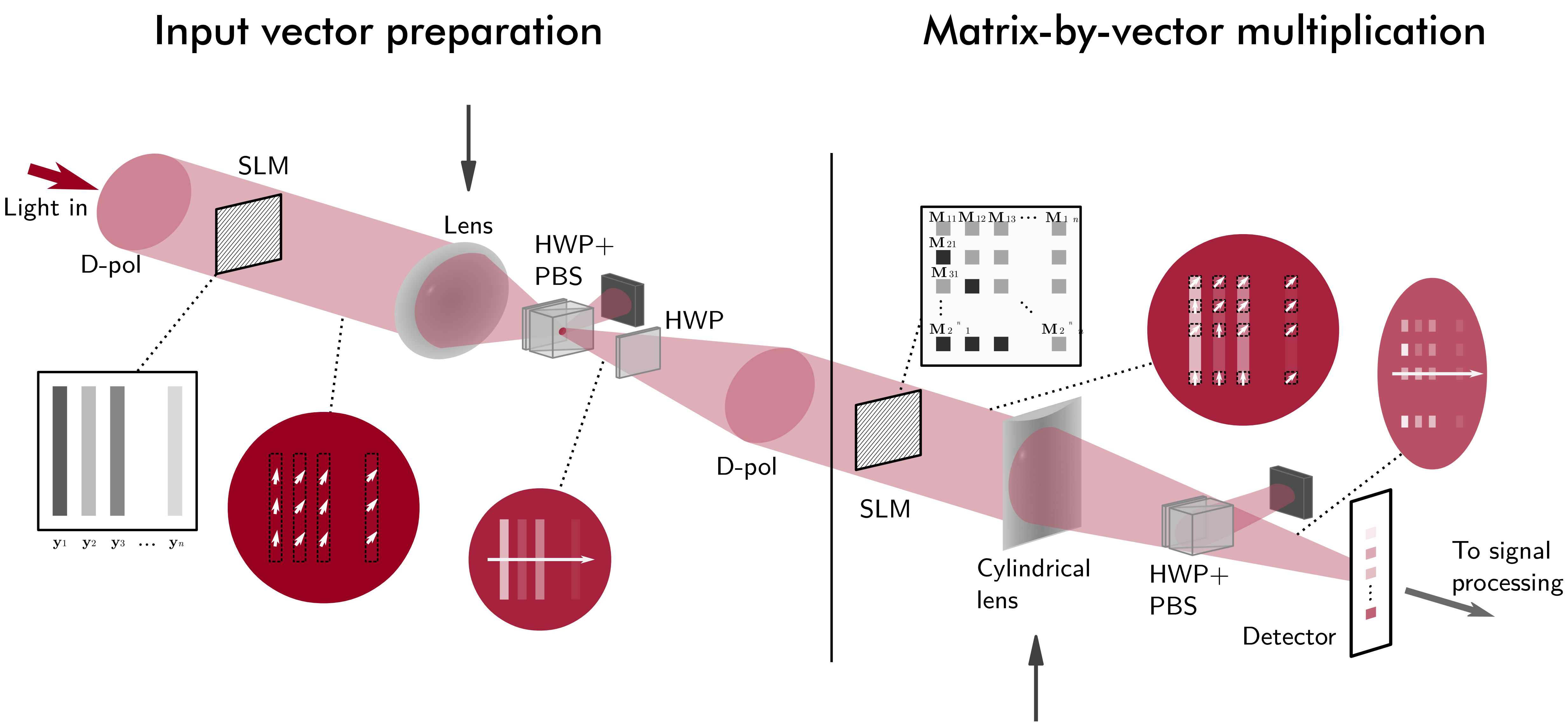


Preparing vectors and matrices

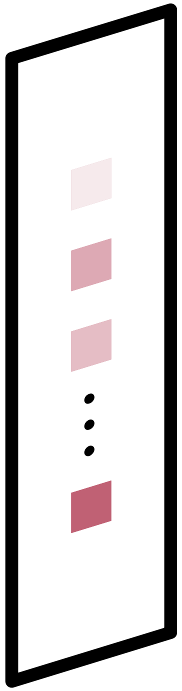


Input vector preparation




Input vector preparation
Matrix-by-vector multiplication
Light in
D-pol
SLM
Lens
HWP+
PBS
HWP
D-pol
SLM
Cylindrical
lens
HWP+
PBS
Detector
To signal
processing

Output signal



Detected signal

Realizations vector:

$$\mathbf{r} = \mathbf{M} \times \mathbf{y}$$

Feasible solutions

- presence of pixel with a reference intensity value
- corresponding to $x_1 = 0$

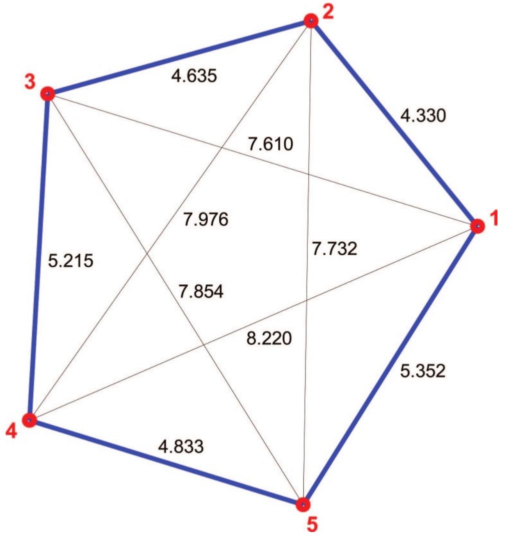
Discussion and Outlook

Performance

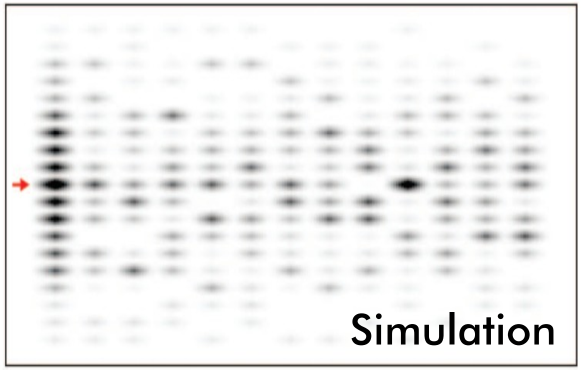
- A figure of merit – MAC-operations/s
  - **Here**: 0.1 TOPS
  - **Photonic chip**: 2 TOPS [1]
  - **CPU**: 2.6 TOPS [2]
- Possible limitations
  - Noise level
  - Refreshing rate of SLM
  - SLM resolution
- Speeding up
  - Larger / faster SLM
  - Many SLMs in parallel
  - Alternative modulation scheme

NP-hard!

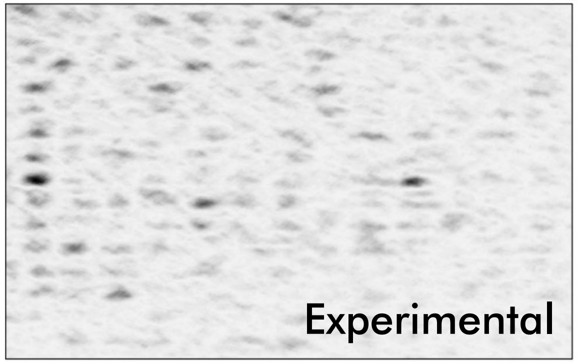
Other optical schemes



Traveling Salesman Problem [3]



Simulation



Experimental

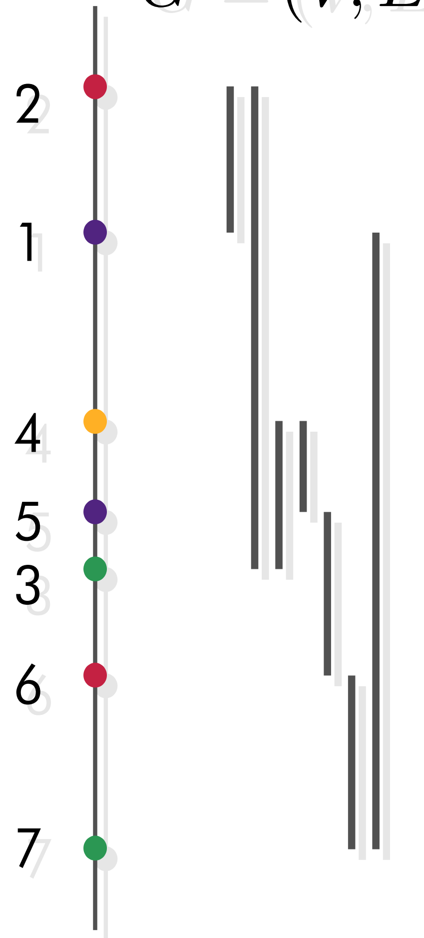
[1] Feldmann *et al.*, Nature **589** (2021)

[2] Jouppi *et al.*, Proc. ISCA '17 (2017)

[3] Shaked *et al.*, Appl. Optics **46** (2007)

Summary

$$G = (V, E, d)$$



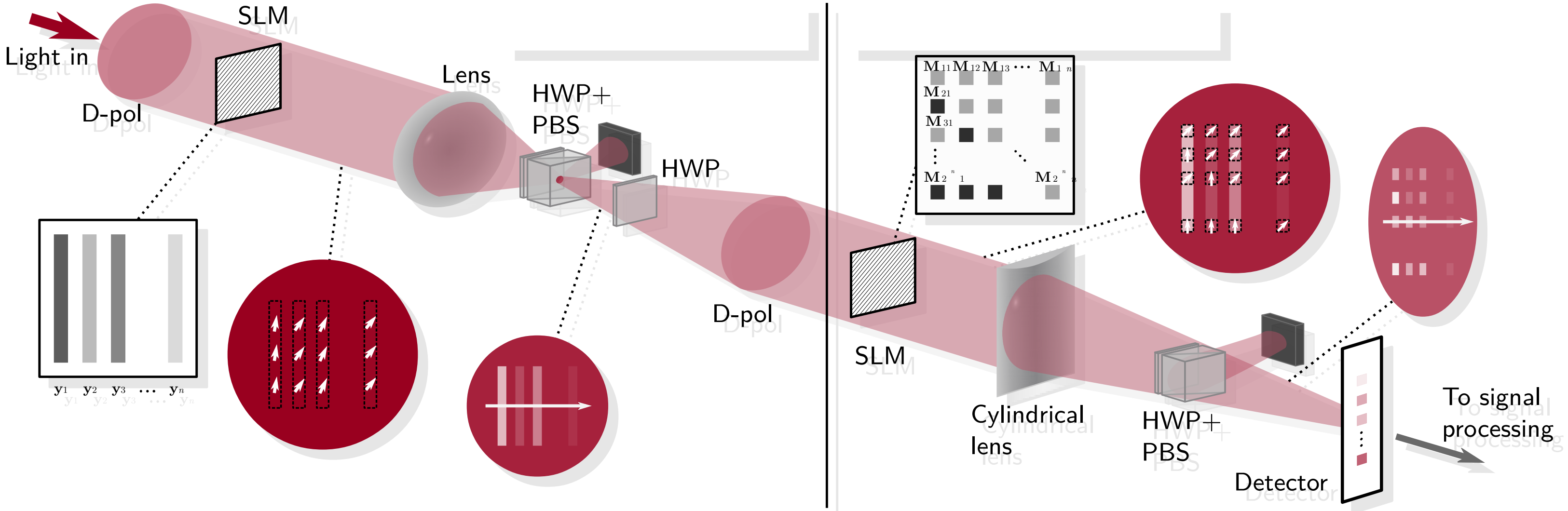
Paradoxical DGP_1

$$\mathbf{r} = \mathbf{M} \times \mathbf{y}$$

$$d \rightarrow \mathbf{y}$$

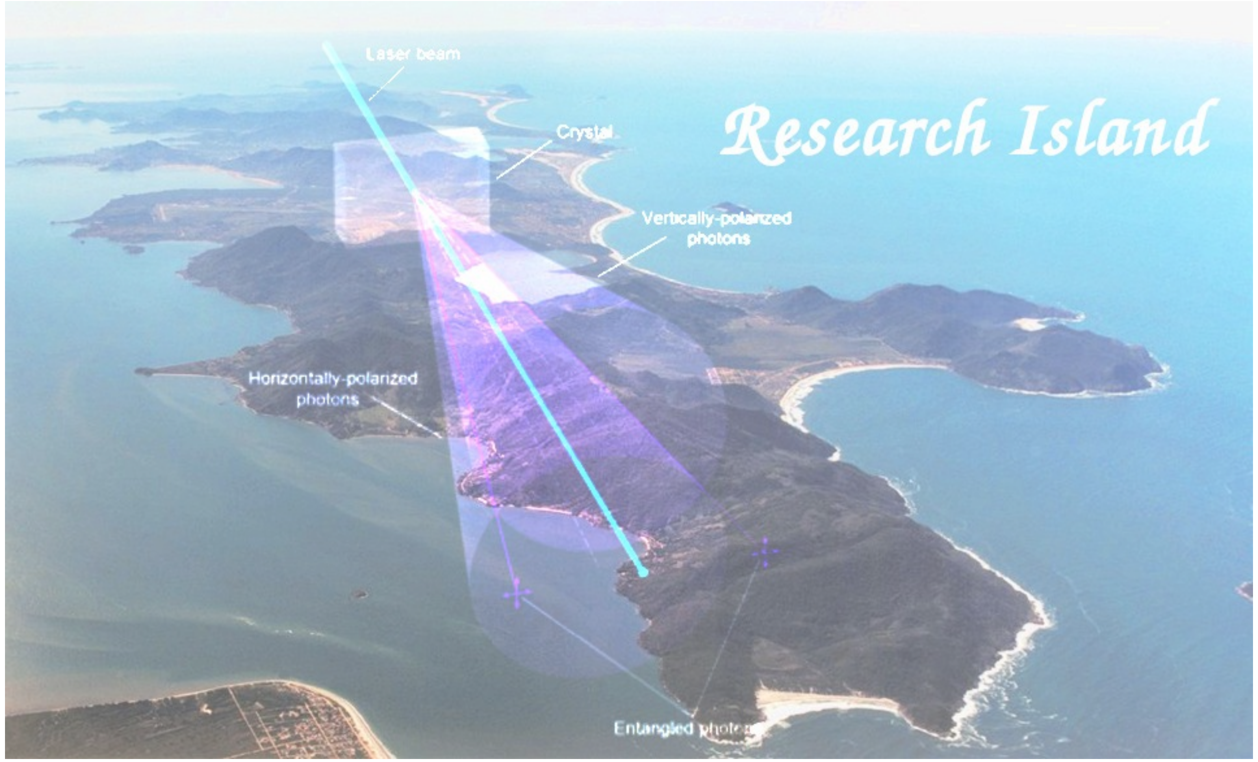
$$\mathbf{M} = \begin{pmatrix} -1 & -1 & -1 \\ 1 & -1 & -1 \\ -1 & 1 & -1 \\ 1 & 1 & -1 \\ -1 & -1 & 1 \\ 1 & -1 & 1 \\ -1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

Matrix-by-vector reformulation allows for...



...Optical computing of matrix-by-vector multiplication

Acknowledgements



Thank you for your attention!

INCT-IQ
Instituto Nacional de Ciência e Tecnologia de Informação Quântica

UFSC

CNPq
Conselho Nacional de Desenvolvimento Científico e Tecnológico

fapesc
Fundação de Amparo à Pesquisa e Inovação do Estado de Santa Catarina

CAPES

AGENCE NATIONALE DE LA RECHERCHE
ANR

UNIVERSITÉ DE RENNES 1