VSB TECHNICAL UNIVERSITY OF OSTRAVA | IT4INNOVATIONS NATIONAL SUPERCOMPUTING CENTER



GBS4EO

Gaussian Boson Sampling for Earth Observation

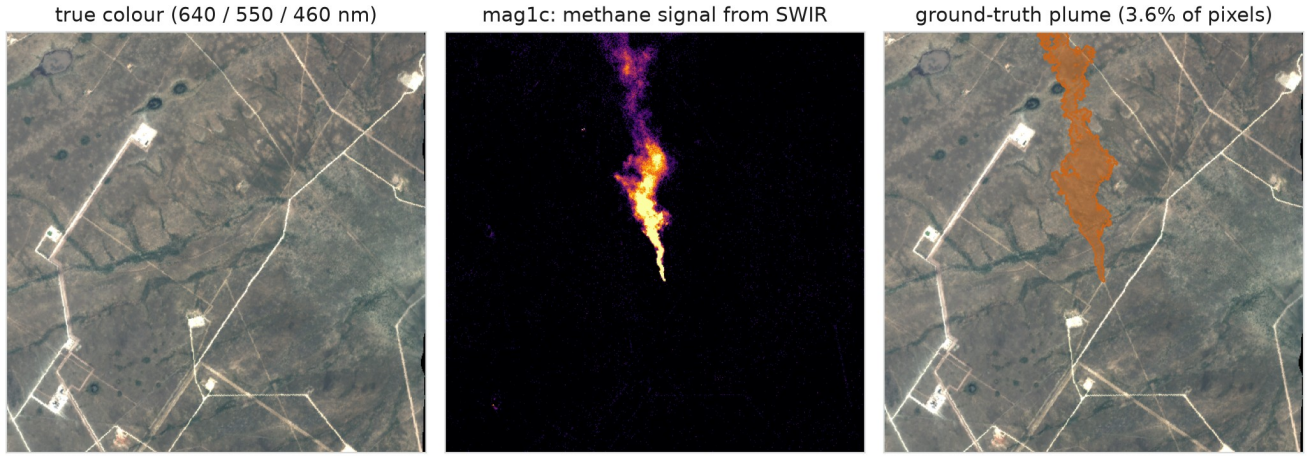
Qiskit Fall Fest 2026

Michal Belina | IT4Innovations

Outline

- Motivation
  - Methane detection – classification
  - HYPERVIEW experiment – band selection
- Principle of Gaussian Boson Sampling
- Algorithm description
- Results
- Hardware requirements

Why Earth-observation data are hard: the methane case



- Invisible to the eye: methane shows up only as weak absorption in the short-wave infrared
- Extreme imbalance: a detector must find tiny, irregular plumes in mostly empty scenes
- Few, costly labels; plume shape changes with wind and source

0.21%

of all pixels are plume (342 STARCOP test tiles)

51%

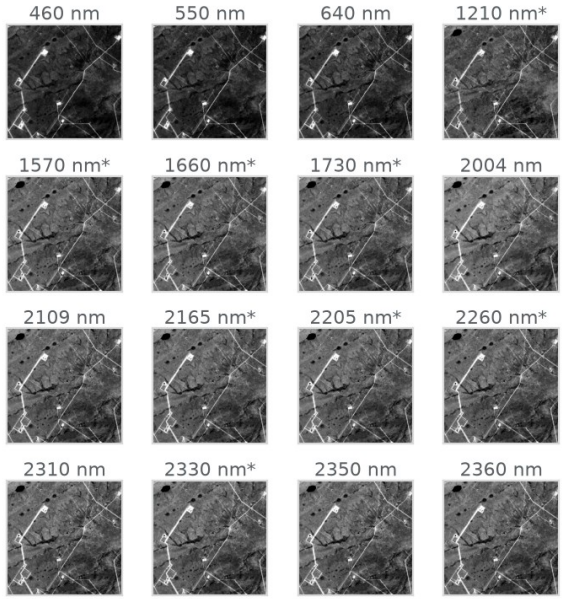
of the tiles contain no plume at all

≈ 425 bands

per pixel (AVIRIS-NG): high-dimensional, redundant

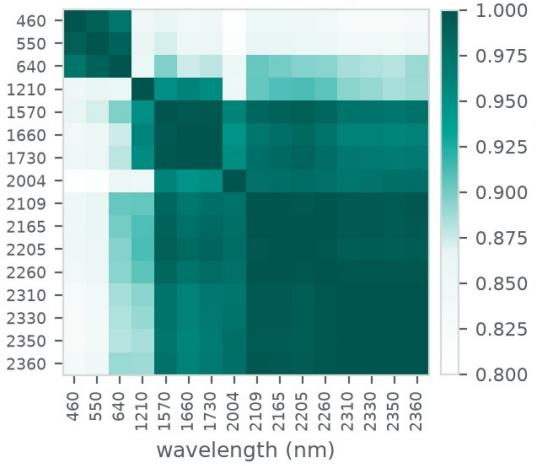
Band redundancy

16 bands of one tile, each on its own contrast scale

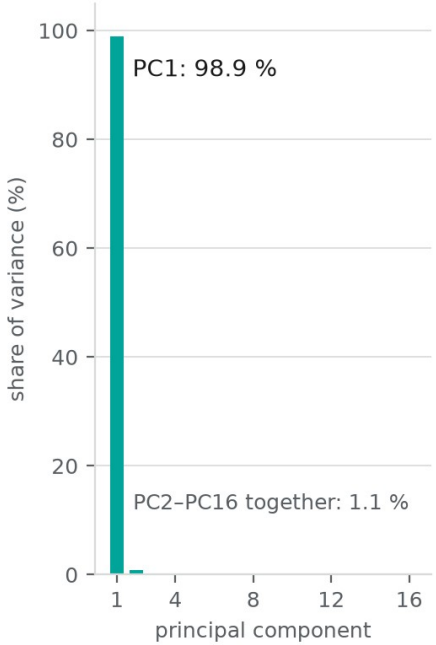


* WorldView-3 SWIR band; the others are AVIRIS-NG

band-to-band correlation (min 0.81)



one direction carries almost everything

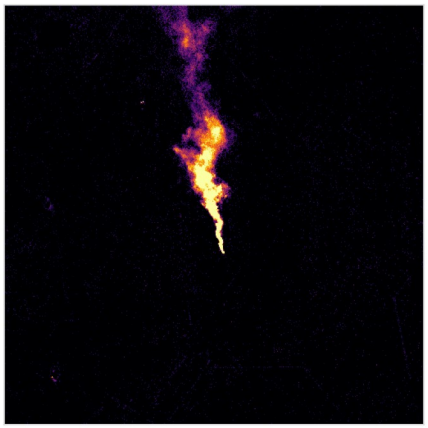


16 bands, one story: every pair correlates at ≥ 0.81, one component holds 98.9 % of the variance

Classification – methane

Which pixel contains a methane plume?

mag1c: methane signal from SWIR



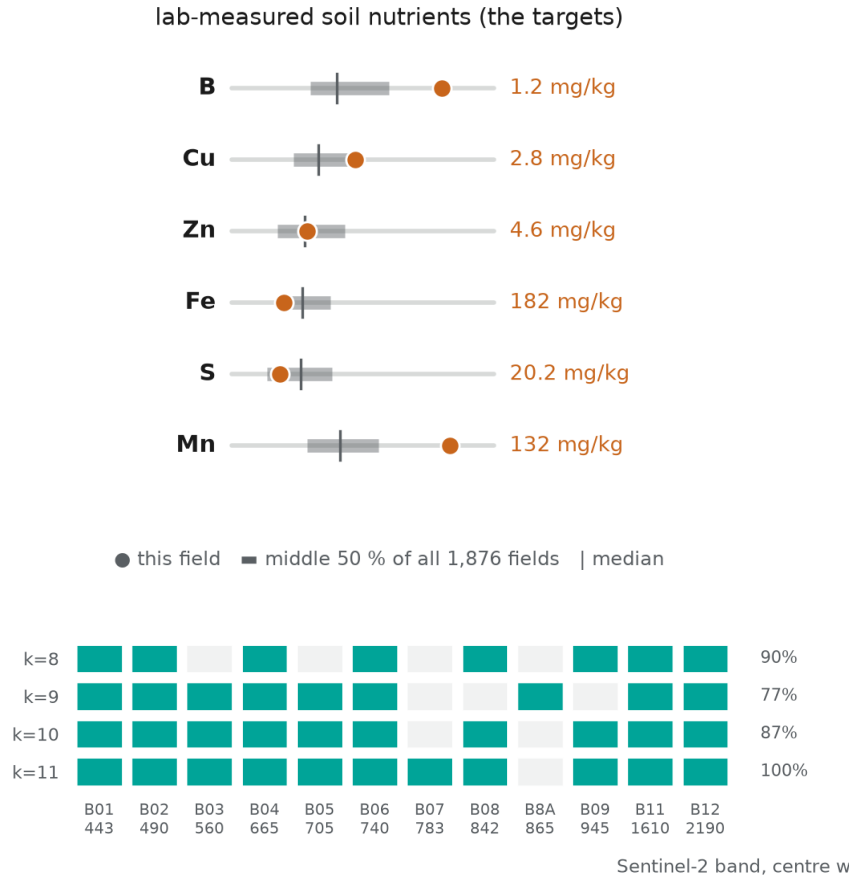
ground-truth plume (3.6% of pixels)



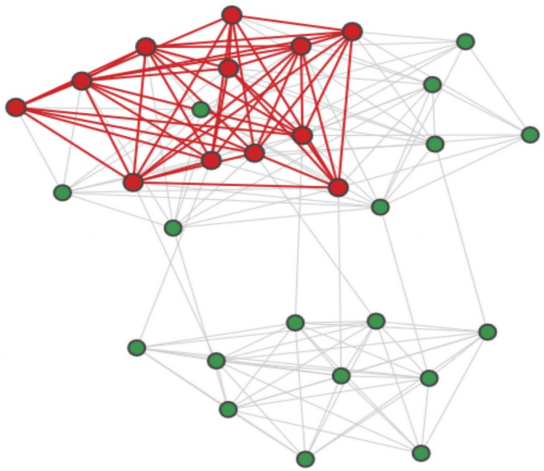
- Per pixel: plume or background
- STARCOP: only 0.21 % of pixels are plume
- Score: F1 of the methane mask

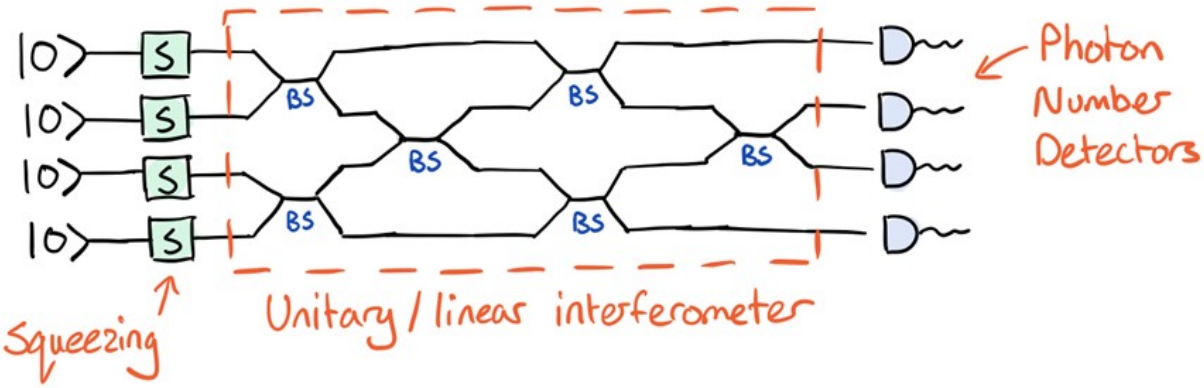
Band selection – HYPERVIEW

Which bands can be omitted from the analysis?



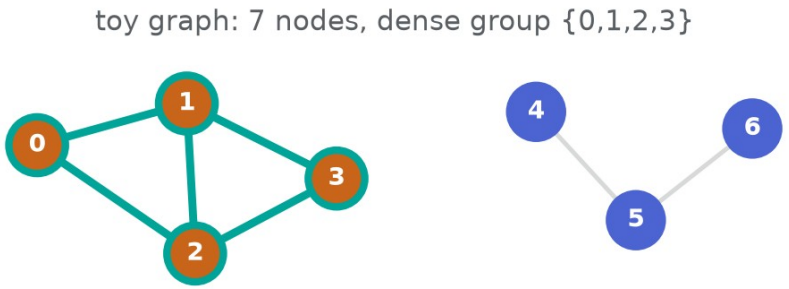
Core idea: GBS prefers dense subgraphs I

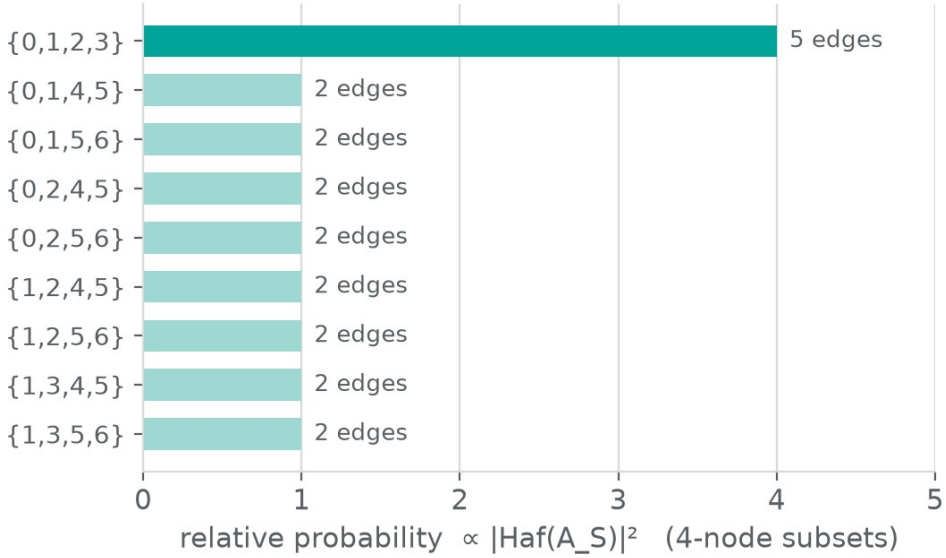




- Squeezed light enters an interferometer; detectors count the photons in each output mode
- One mode = one graph node, so each click pattern selects a node subset
- Dense subgraphs (red) are sampled far more often than sparse ones (Arrazola & Bromley, 2018)

Core idea: GBS prefers dense subgraphs II





- **Encode** the graph's adjacency matrix A in a photonic circuit; each detected photon pattern selects a node subset S
- **P(S) ∝ |Haf(A_S)|²**: the hafnian counts perfect matchings, so dense subsets are sampled most often. Here {0,1,2,3} is 4× more likely than any other 4-node subset

The Quantula pipeline, end to end



Swappable sampler: one line of code

Piquasso
photonic simulator; all results in this talk

Gate-based
gbs-compiler → QASM → Qiskit or VLQ

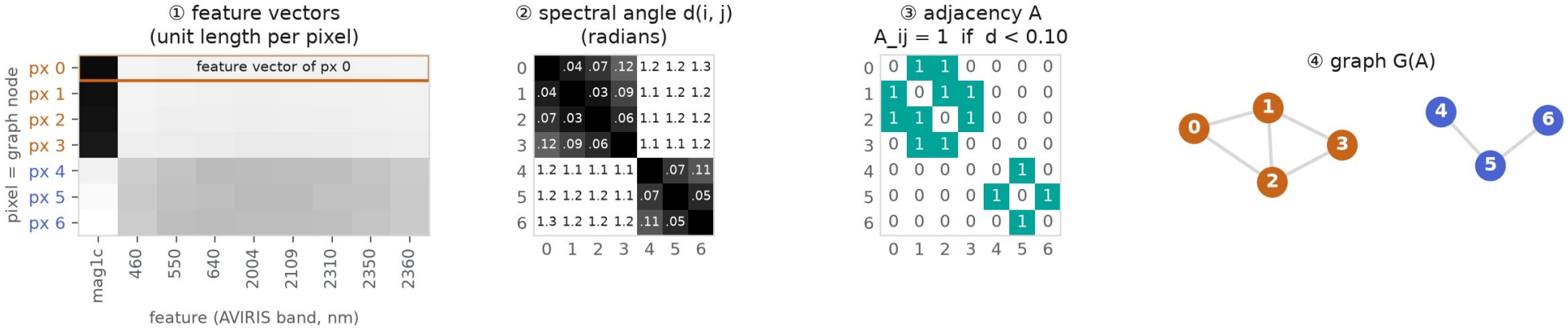
Then a task-specific step

Band selection
disjoint clusters → one band each → regression

Methane detection
anchor vote per graph → pixel mask

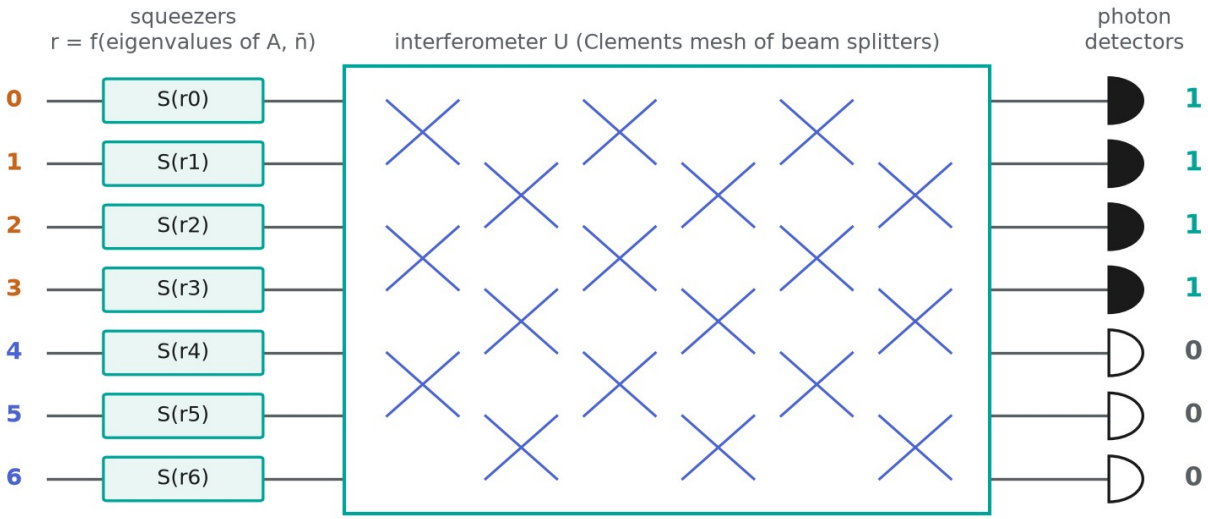
Every sampler call is one circuit execution: on hardware, one queued job (see the VLQ part)

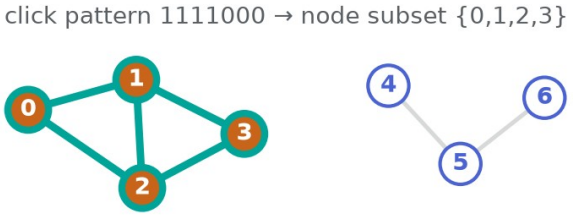
Step 1: from feature vectors to the adjacency matrix



	Band selection (HyperView-2)	Methane detection (STARCOP)
Node	one spectral band	one pixel
Vector	the band's values over all training fields	mag1c + 8 AVIRIS bands, scaled to unit length
Edge if	distance < τ (HPO: metric and τ; best cosine, 0.0022)	spectral angle < 0.10 rad; query pairs ≤ 8 px apart
Graph	12 nodes (12 Sentinel-2 bands)	14 nodes: 6 query + 4 methane + 4 background anchors

Step 2: from the adjacency matrix to a GBS circuit





Encode

Takagi decomposition of A: eigenvalues → squeezers, unitary → beam-splitter mesh. One mode = one node

Set the brightness

squeezing scaled so the circuit carries n̄ photons in total (methane: n̄ = 3.5)

Read out

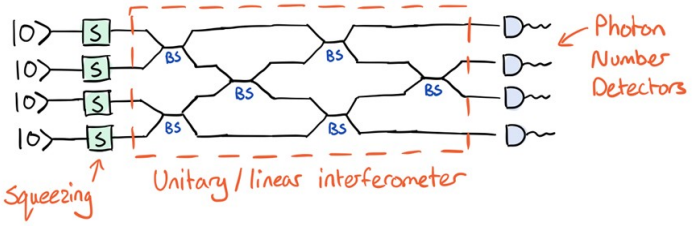
keep shots with ≤ 1 photon per mode; each shot is a node subset

Step 2: from the adjacency matrix to a GBS circuit

PIQUASSO

Source: github.com/Budapest-Quantum-Computing-Group/piquasso

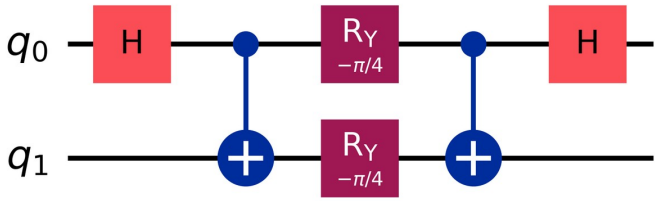
Photonic QC layout

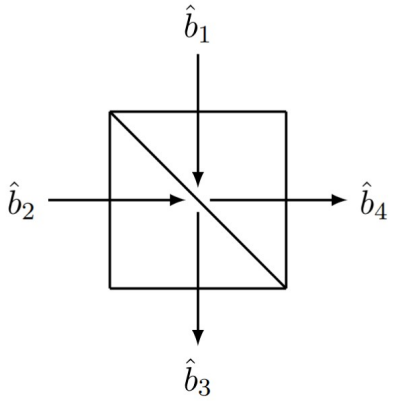


Source: pennylane.ai

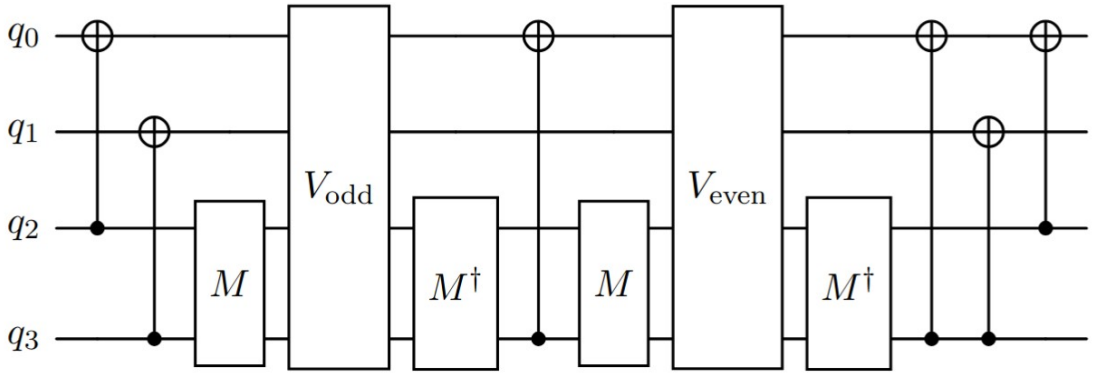
GBS compiler, Jagiellonian University, Kraków

Gate-based QC layout (beam splitter only)

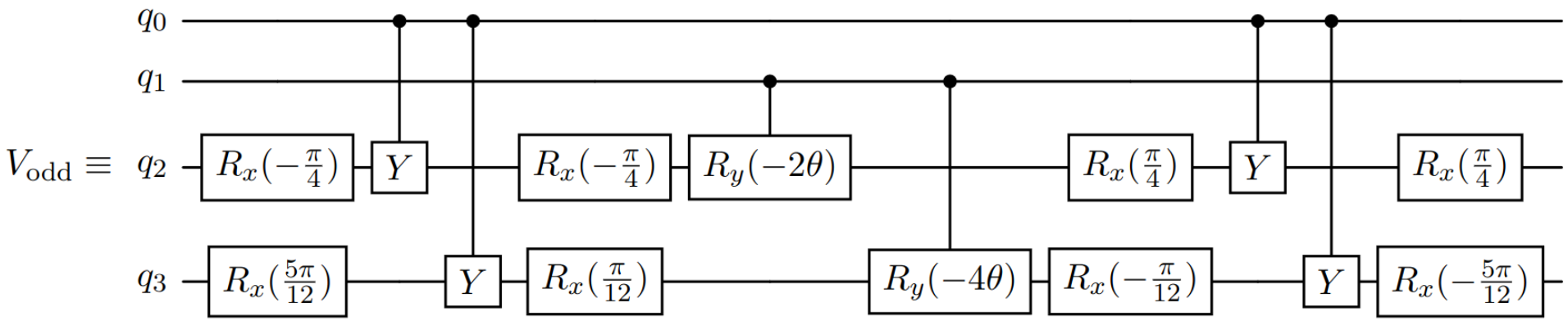


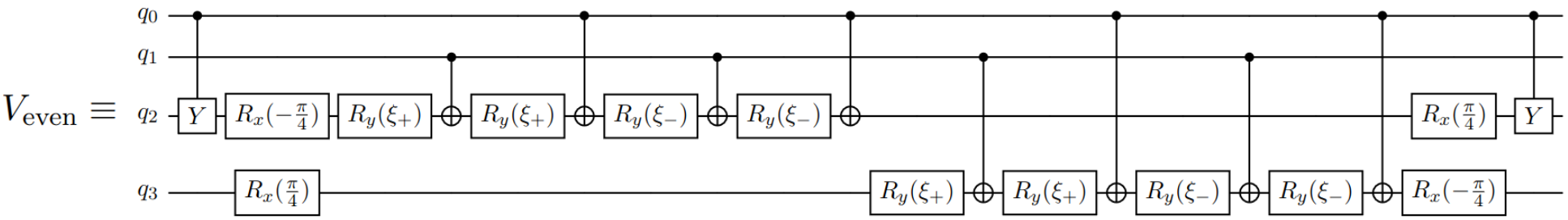


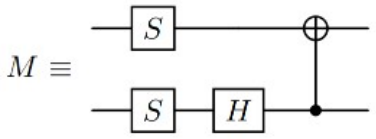
$$\hat{U}_{BS}(\theta, 0) \equiv$$



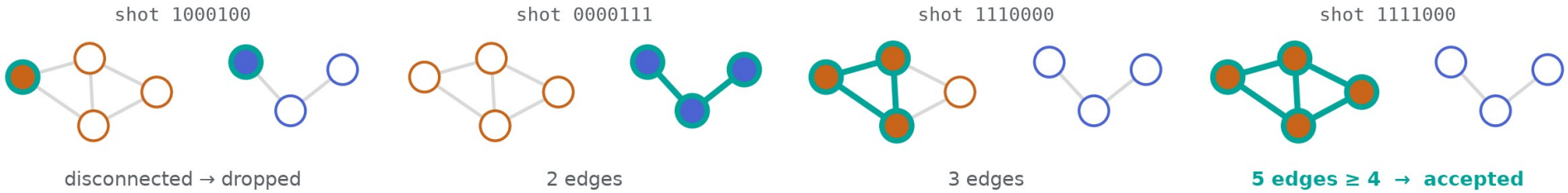
GBS principle: no photonic chip, so we compile to qubits







Step 3: after the run, pick one dense subgraph



1 Sample the same circuit: n shots per iteration (methane: 10)

2 Drop subsets with < 2 nodes, duplicates, or disconnected ones

3 Candidate = the subset with the most edges

4 Accept if edges ≥ $\lfloor \bar{n}(\bar{n}-1)/2 \rfloor - k$, otherwise sample again with the bar lowered (k = iteration)

Acceptance bar, $\bar{n} = 3.5$

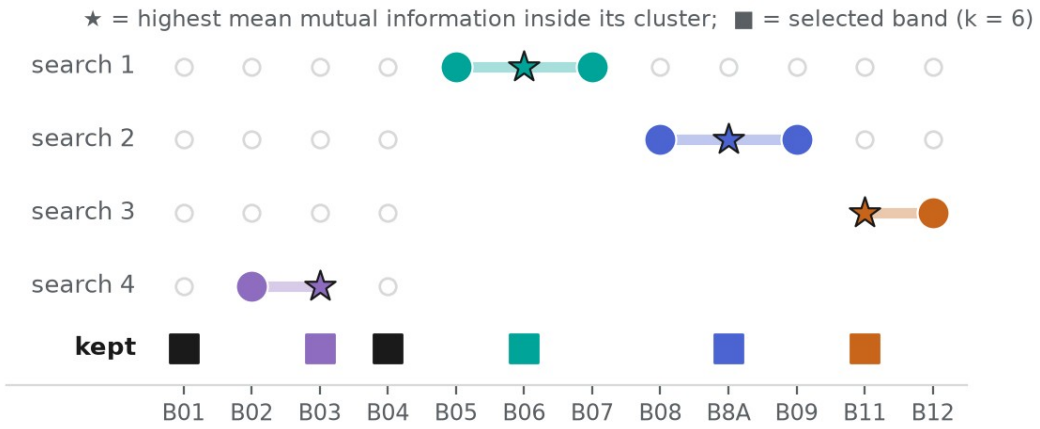
k = 0: ≥ 4 edges

k = 1: ≥ 3 edges

…

$\bar{n}$ sets both the cluster size and the bar

Band selection, from clusters to a band set

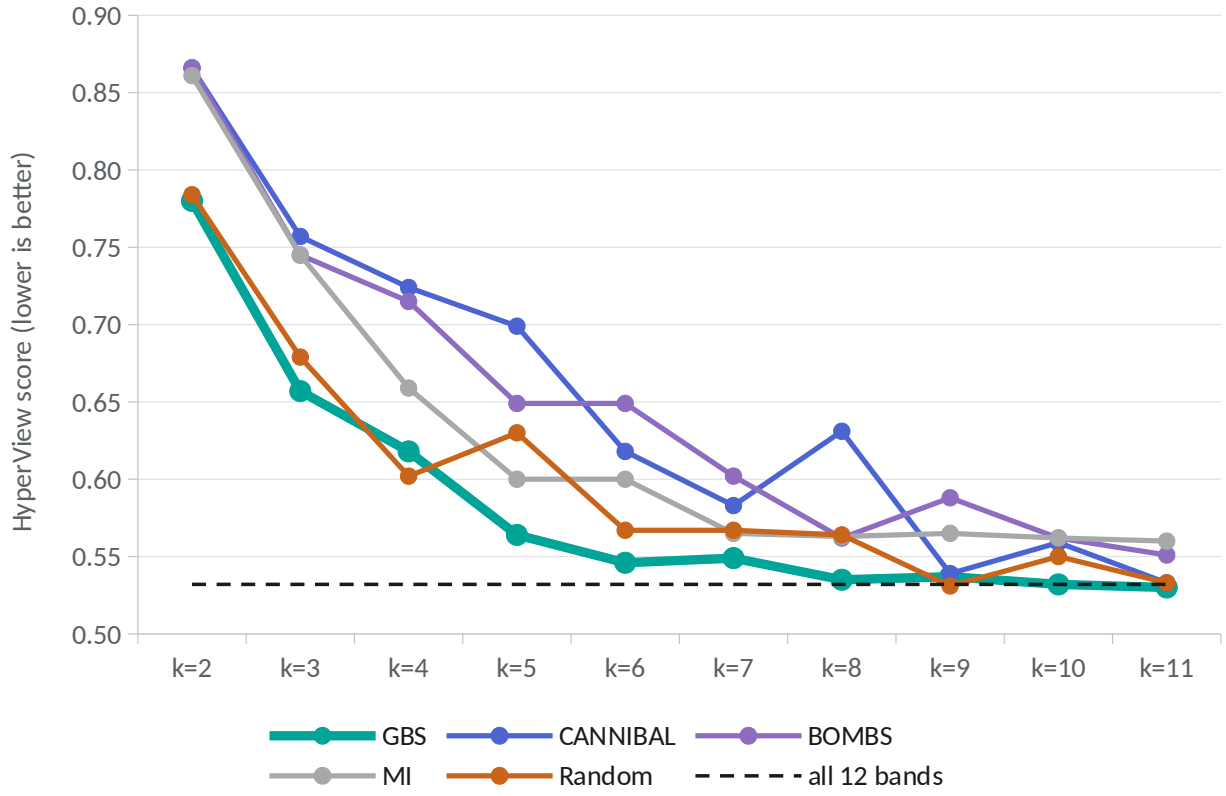


illustration, not a real run

- Accept a dense cluster, remove its bands, search the smaller graph again, until no edges remain
- Clusters are disjoint; leftover bands stay as singletons
- Per cluster, keep the band with the highest mean mutual information with the others (no labels)
- The number of bands k is an outcome, not an input



Band selection: GBS beats the TN1 band sets



8 of 10

values of k where GBS has the lowest score

p = 0.002

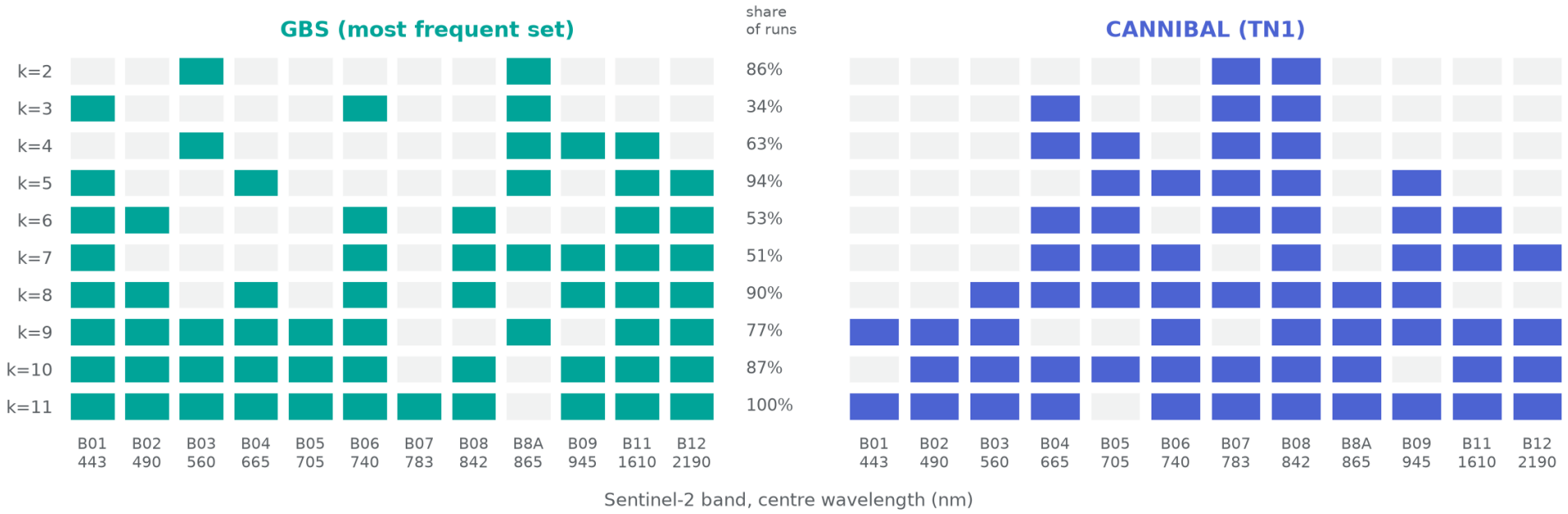
GBS vs CANNIBAL, BOMBS, MI (paired over k = 2-11)

k ≥ 8

GBS matches all 12 bands (0.532)

HyperView-2: 1,876 fields, 12 Sentinel-2 bands → 6 soil nutrients; random forest, 10 random 75/25 splits

Which bands are selected



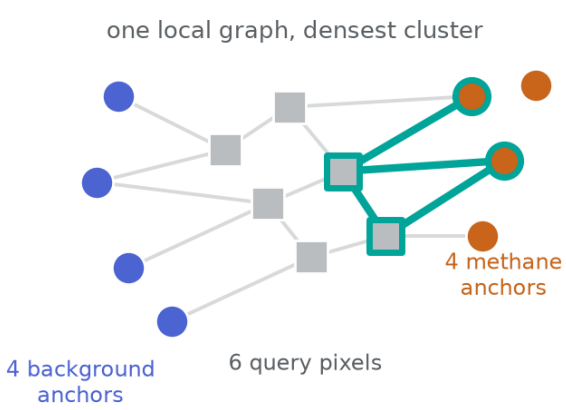
GBS spreads over the spectrum

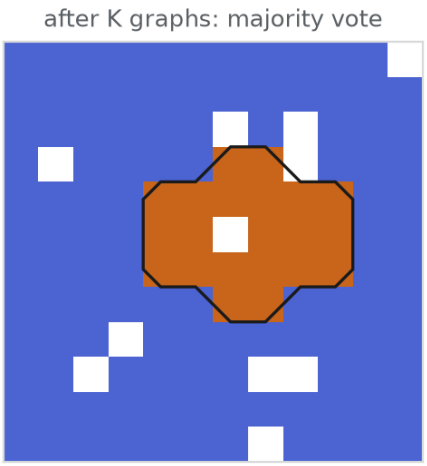
visible, red edge / NIR and SWIR; k = 5: B01 B04 B8A B11 B12 in 94% of runs

CANNIBAL clusters in red edge / NIR

neighbouring bands, e.g. k = 2: B07 B08

Methane detection, from clusters to a mask





- Each graph: 6 query pixels from a 9 × 9 window + 4 methane and 4 background anchors from training data
- One cluster per graph: 10 shots, ≤ 2 iterations
- Only methane anchors in the cluster → its query pixels get a methane vote; only background → background vote; mixed → none
- Majority vote per pixel; never-voted pixels count as background

illustration, not a real run

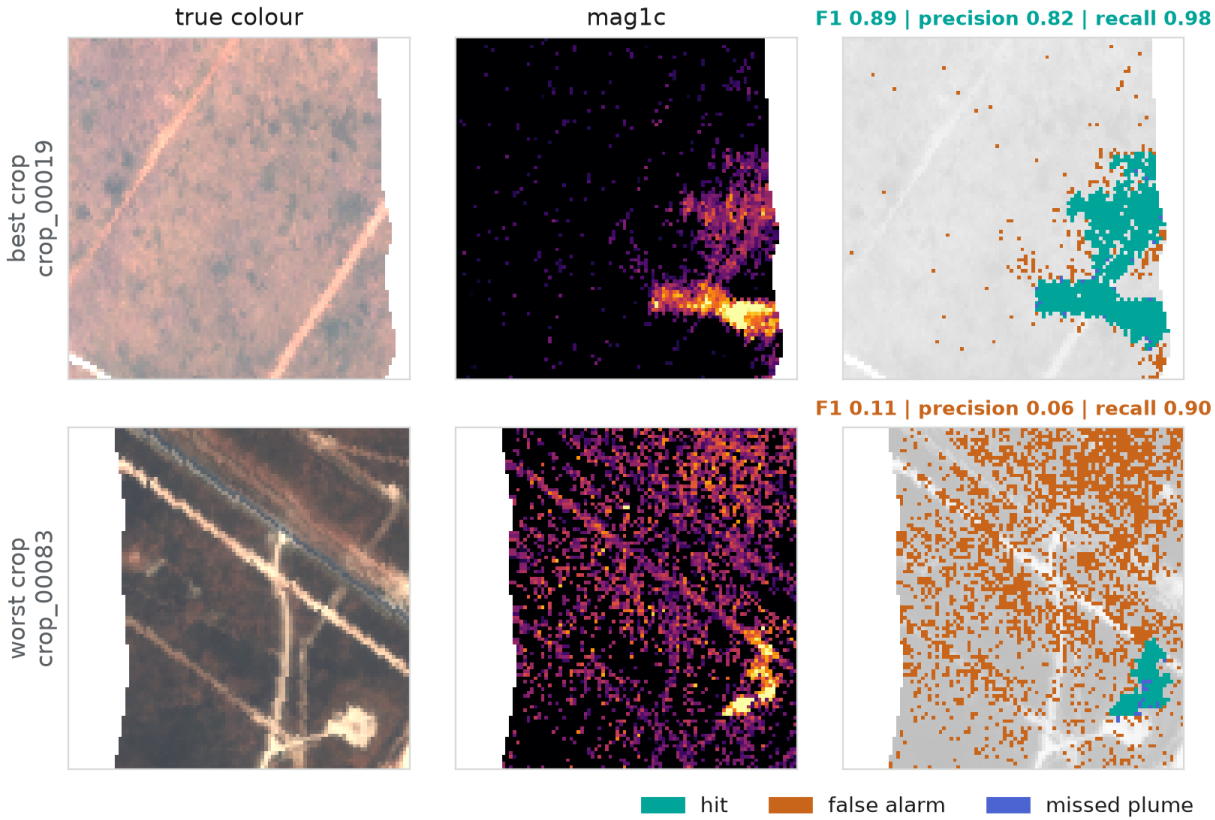
16,000
graphs per 96 × 96 crop (K)

≈ 10
query visits per pixel

16,000+
sampler calls per crop

$$F_1 = \frac{2}{\mathrm{recall}^{-1} + \mathrm{precision}^{-1}} = 2\frac{\mathrm{precision} \cdot \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}} = \frac{2\mathrm{TP}}{2\mathrm{TP} + \mathrm{FP} + \mathrm{FN}}$$

What a detection looks like



Best: F1 0.89

plume shape recovered, few scattered false alarms

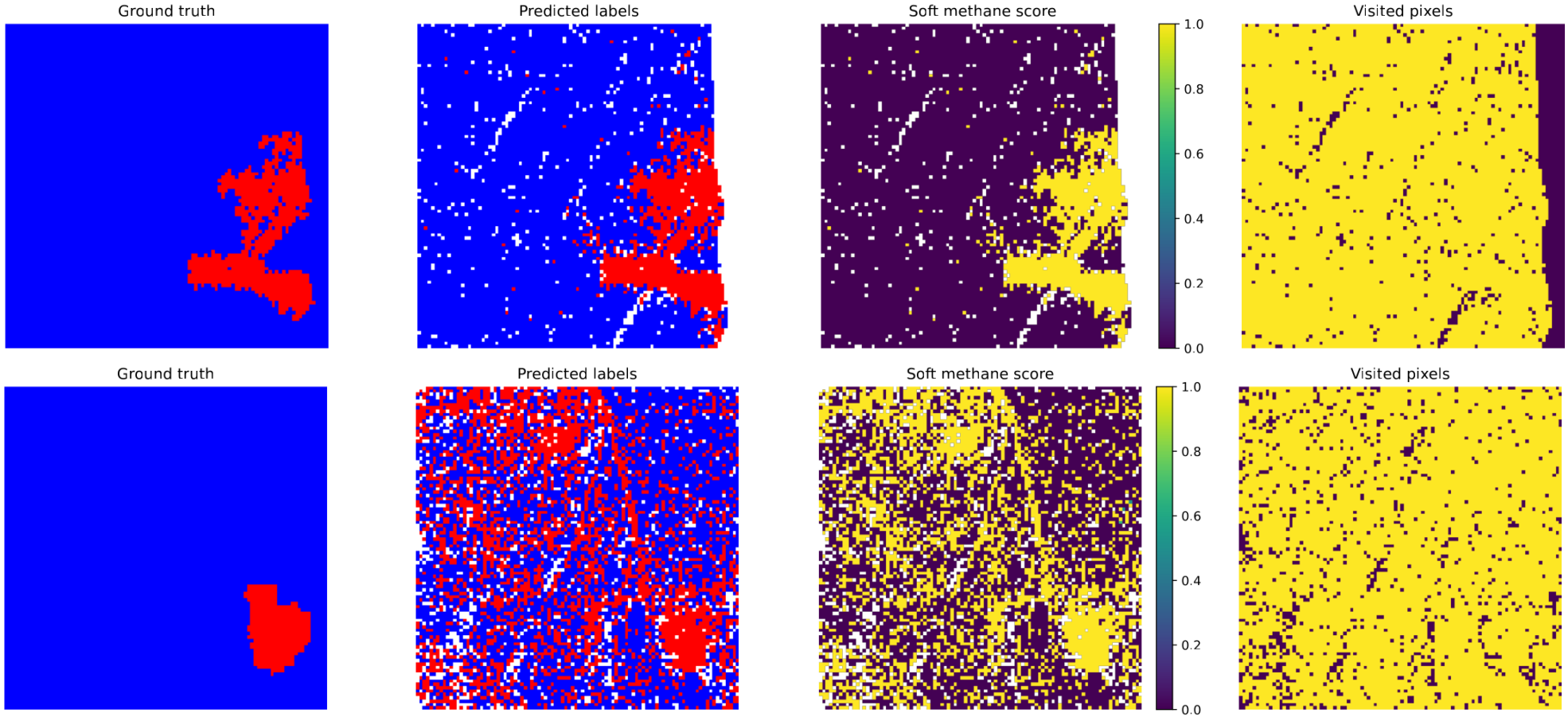
0.629

mean F1 on 100 test crops

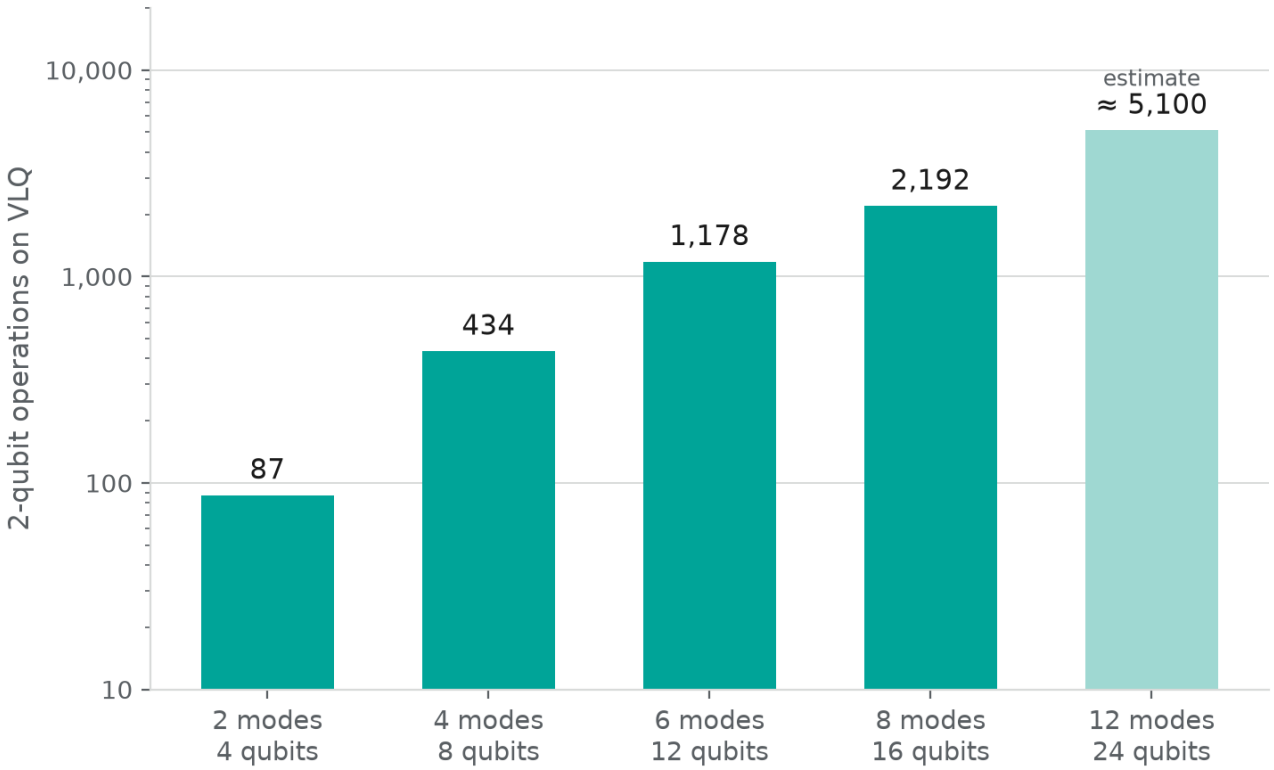
Worst: F1 0.11

mag1c noisy over the scene: roads and bright soil look like methane

What a detection looks like



From photons to qubits: circuits grow fast



2 qubits per optical mode

photon number cut at 3 (Fock cut-off 4)

Star topology

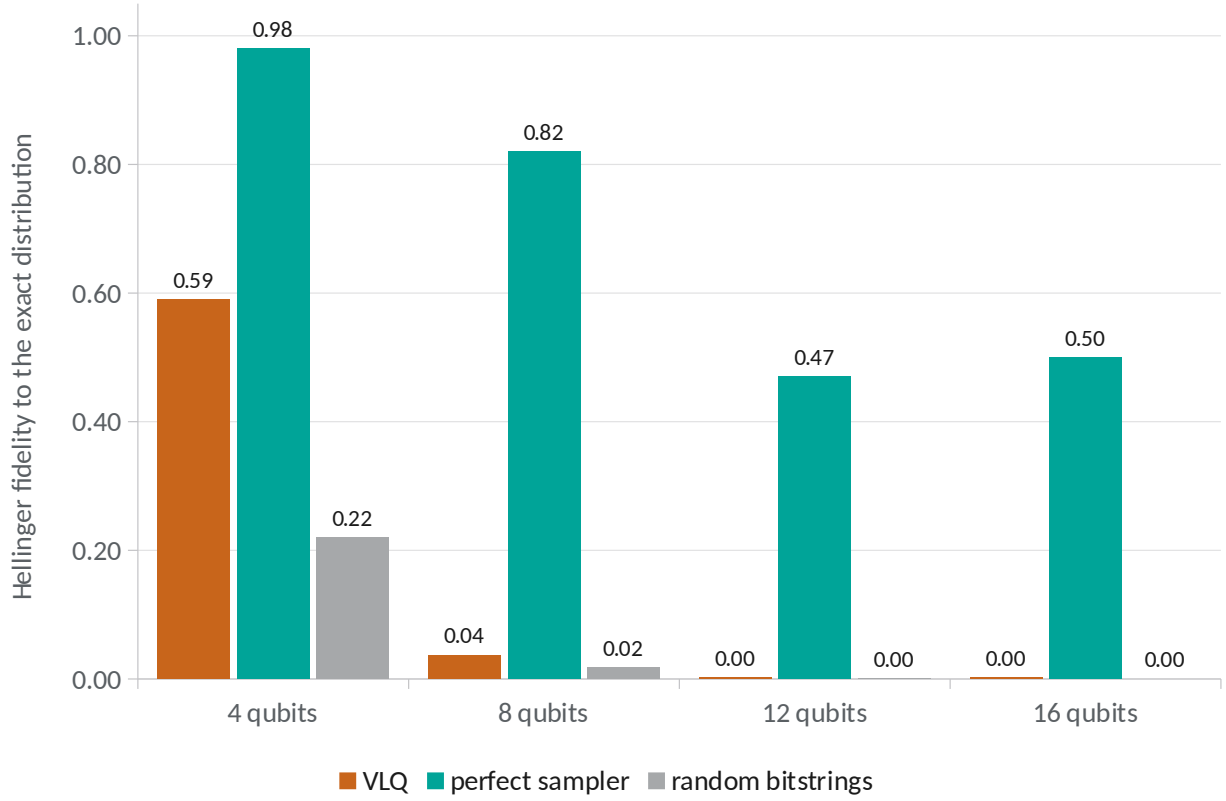
every 2-qubit gate goes through the central resonator, one after another

Real graphs

HyperView, 12 bands: 24 qubits = all of VLQ, ≈ 5,100 ops (estimate)

methane, 14 nodes: 28 qubits, does not fit

On the hardware: only the smallest circuit keeps a signal



0.59

fidelity at 4 qubits (87 two-qubit operations)

≥ 8 qubits

output indistinguishable from random bitstrings

no gain

from Qiskit resynthesis: the compiler's blocks are already cheaper

Verdict: what runs on VLQ today

Works

- ✓ 2-mode circuit (4 qubits, 87 two-qubit operations): fidelity 0.59
- ✓ End-to-end path: compile → submit via LEXIS → decode samples
- ✓ Noiseless gate-based simulation up to 24 qubits (≈ 1 min per call, estimate)

Out of reach

- ✗ ≥ 8 qubits: the output is noise
- ✗ Real graphs: 12 bands ≈ 5,100 operations; methane graphs need 28 qubits

Ways forward

Photonic hardware
run GBS natively

Batch iterations
same circuit → one submission

Noiseless simulation
validate the gate-based path up to 24 qubits

Fewer gates
smaller graphs, tighter Fock cut-off, approximate compilation

Summary and next steps (all results from the photonic simulator)

Band selection

GBS beats CANNIBAL, BOMBS and MI at k = 2-10 and matches all 12 bands from k = 8

Methane detection

Tuned GBS segmentation: F1 0.63 on unseen crops

Gate-based (VLQ)

Works end to end, but only the 4-qubit circuit keeps a signal; real graphs need better hardware or encodings

Next steps

- Smarter truncation of the photon-number (Fock) space of the optical circuit
- Approximate compilation of the optical circuit
- Use a noisy simulator to estimate the gate quality needed to run the circuits
- Test other hardware: IBM Quantum devices, the Polish PIAST-Q

Thanks to the people behind the project



Jakub Mielczarek • 1st
Theoretical physicist looking into the future
3d • Edited •

I'm happy to share our first results from the **European Space Agency - ESA** (Phi-lab) funded Gaussian Boson Sampling for Earth Observation (GBS4EO) project. The article and the associated GitHub repository provide theoretical background & Python code of a photonic to digital quantum circuits compiler (COPHI), enabling, e.g., representing GBS on qubit-based architectures.

arXiv: **https://lnkd.in/ePDF8nCz**

GitHub: **https://lnkd.in/ejFJZGDQ**

The project is conducted in the consortium of: **KP Labs**, **Jagiellonian University** and **IT4Innovations National Supercomputing Center**

Many thanks for the cooperation to: **Grzegorz Czelusta**, **Marcin Markiewicz**, **Artur Miroszewski**, **Gabriele Meoni**, Piotr Gawron, **Marek Lampart**, **Jakub Nalepa**, **Tomasz Rybotycki**, **Michal Belina**, **Jiri Tomcala, Ph.D.**., Paweł Skorupa

COΦ – COmpiling PHotonics into dIgital quantum circuits

Thank you

Questions and discussion

Michal Belina, michal.belina@vsb.cz, IT4Innovations National Supercomputing Center