Geometry is the harness of 3D learning

Paul-Edouard Sarlin

Workshop on Visual Localization and Mapping @ ECCV 2026
2026-09-08

From the workshop page:

“Are classical SLAM pipelines becoming obsolete?”

Yes! We love e2e reconstruction models

Robustness – more powerful priors

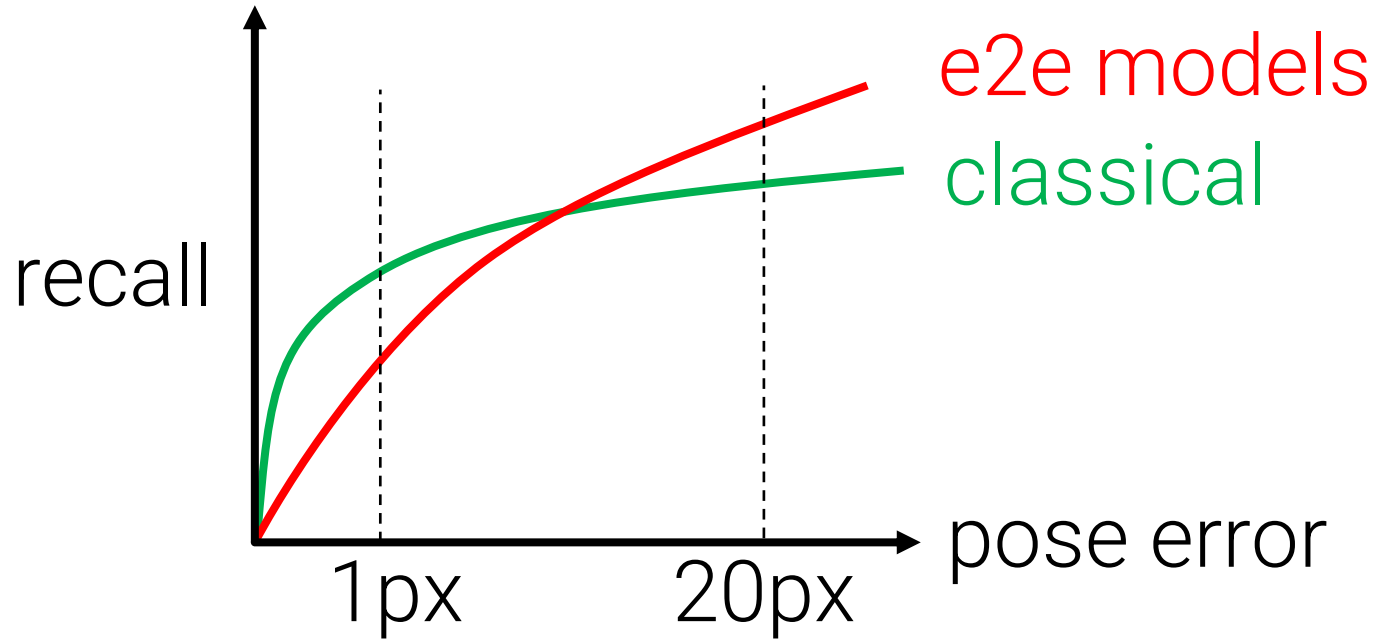
Simplicity – reduced engineering, “only” a Transformer

Efficiency – matmul on GPU go brrr

Pretty results – dense 3D point clouds

Emergent **semantics**…

But they fall short of higher precision



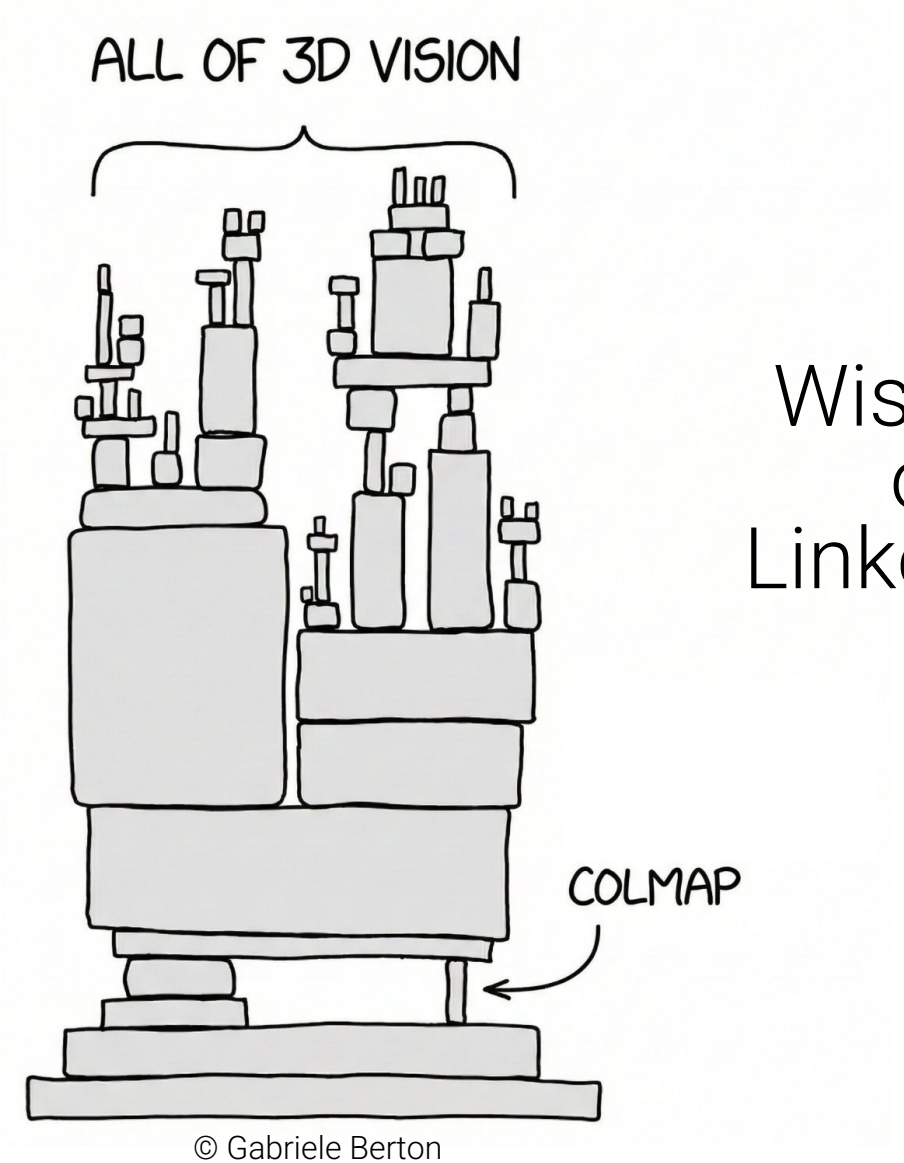
What matters for many **real applications** like 3DGS

What matters for **pretty visualizations** on social media

What can we do about this?

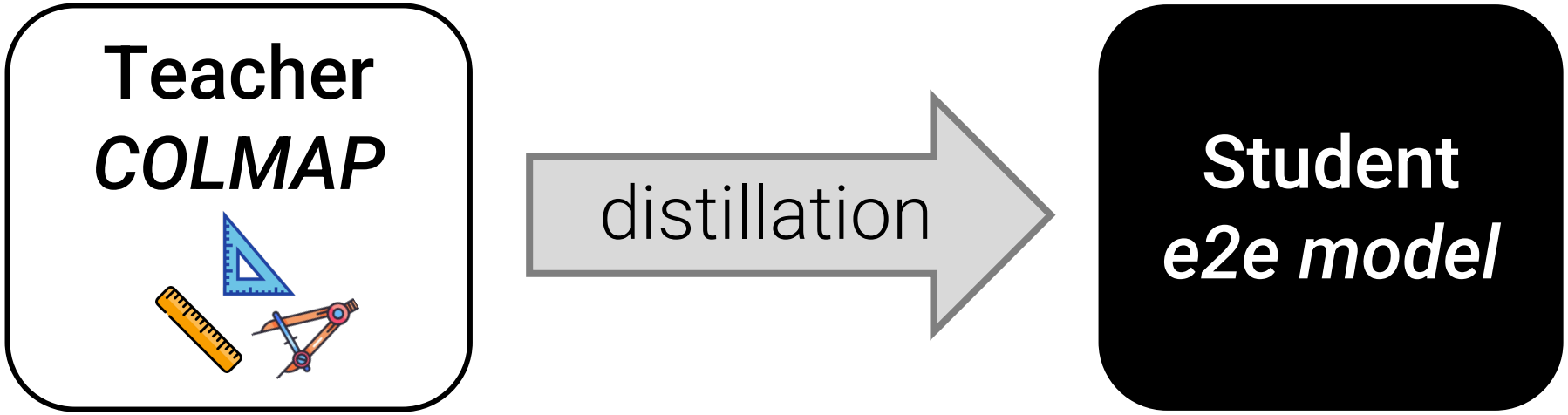
1. Improve the data
2. Improve the model design

Where does the training data come from?



Wisdom of LinkedIn…

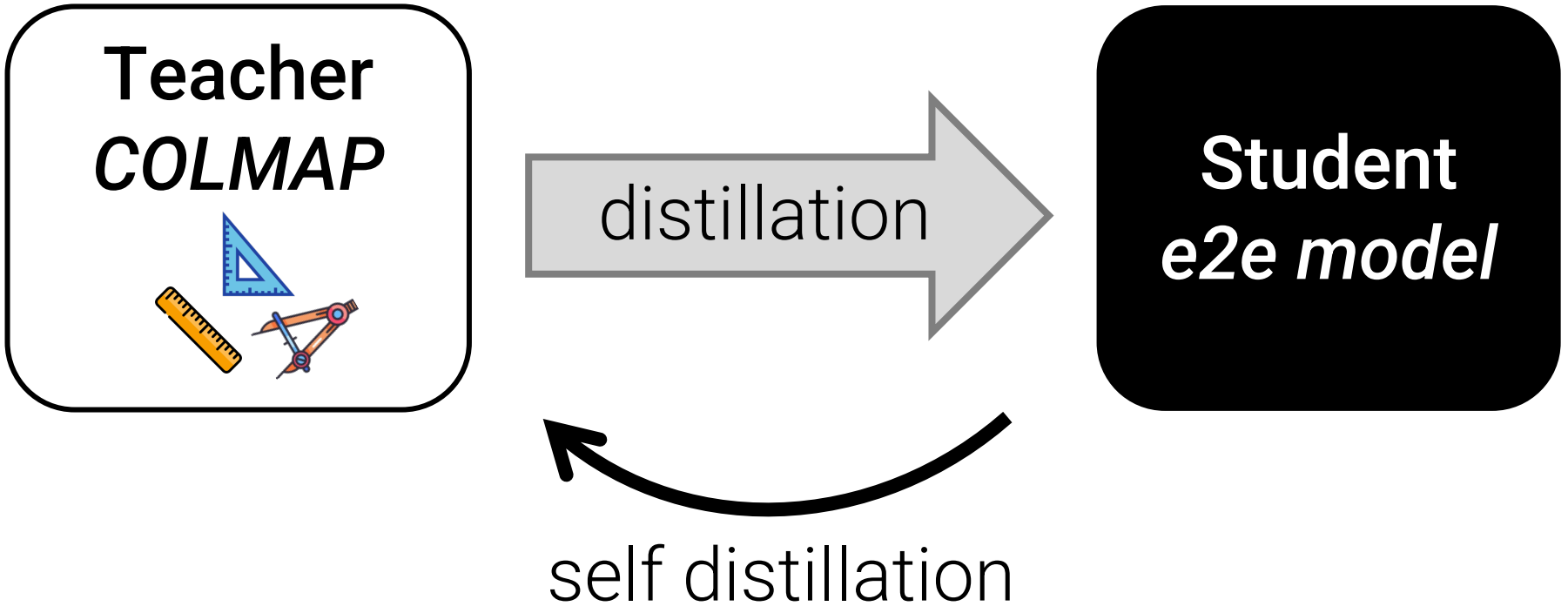
It’s all about viewpoints



Why does this work well?

- **Accuracy**: COLMAP often works well
- **Interpretability**: well-understood metrics quantify the uncertainty
- **Reliability**: we know when it fails

It’s all about viewpoints

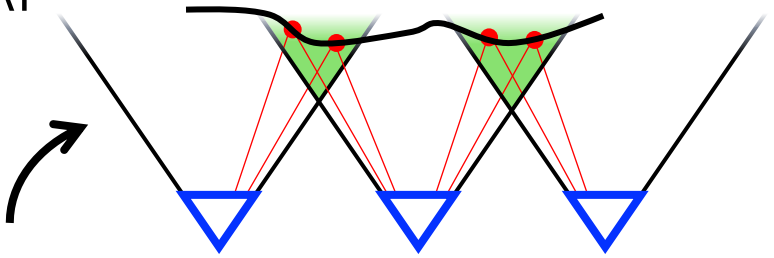


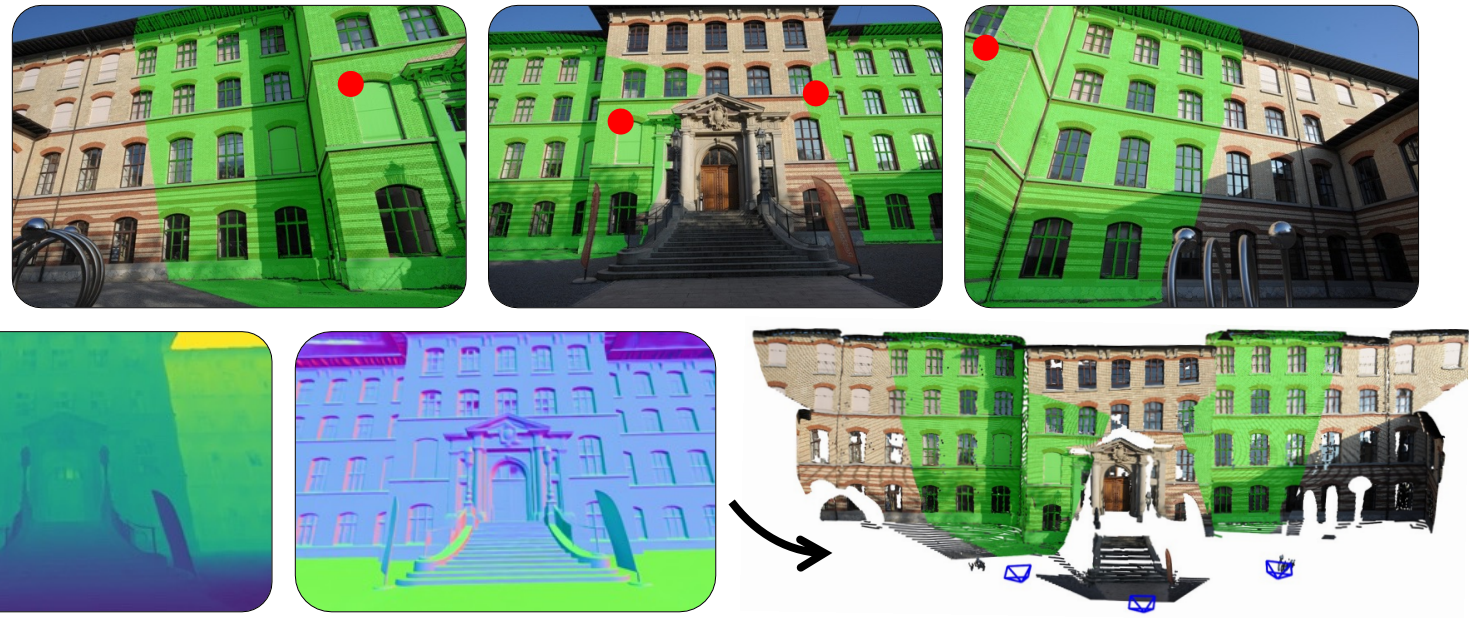
Can we use the student to improve the teacher to improve the student?

“bootstrapping”

MP-SfM: SfM with monocular priors [CVPR'25]

- **Tight** integration of monocular depth, normals, matching
- **Robustness**: ~e2e models, >> COLMAP
- **Accuracy**: >> e2e models, ~COLMAP
- Excels at low overlap

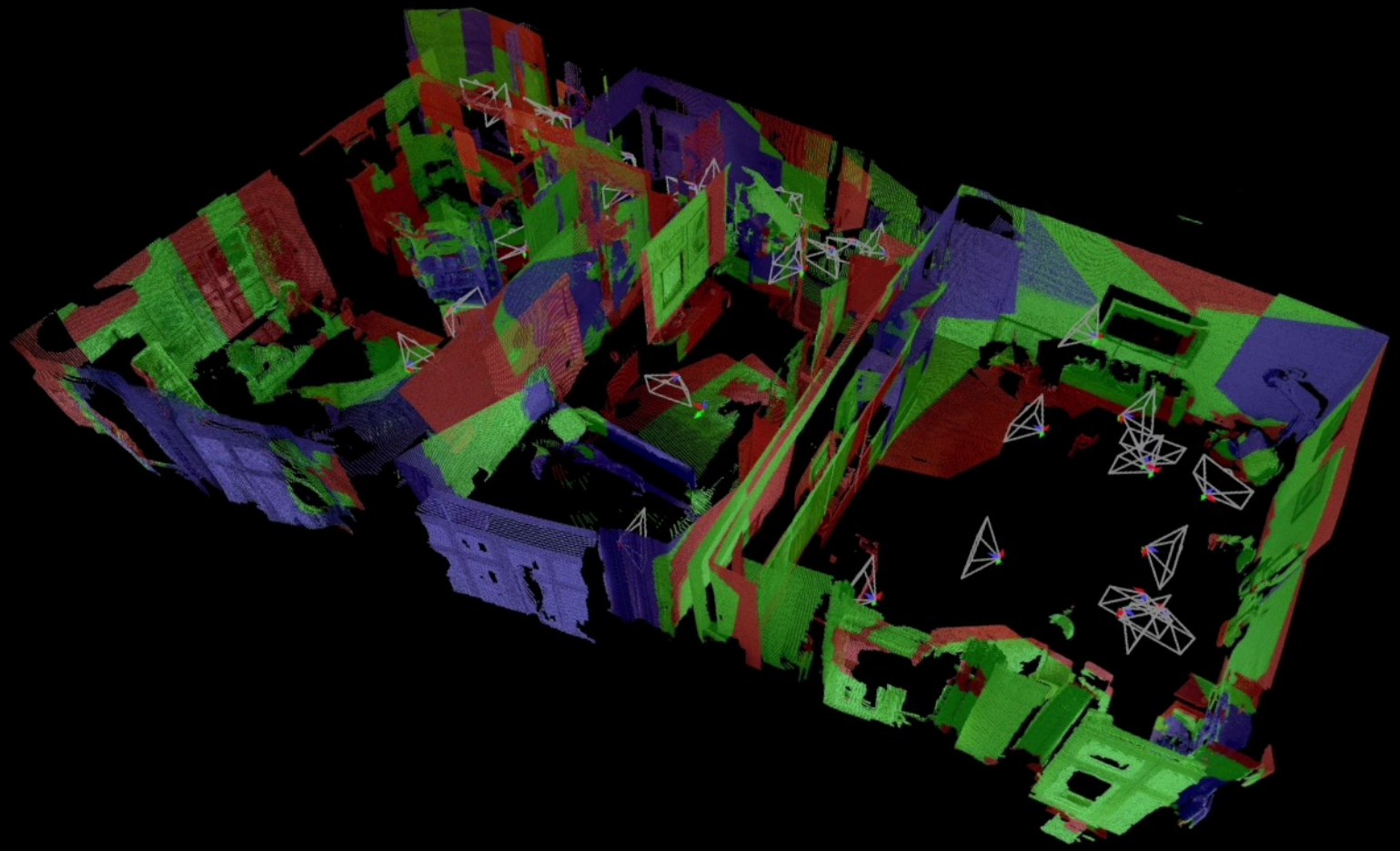




Zador Pataki

Marc Pollefeys

MP-SfM (Ours)



Robust Reconstruction in Extreme Low Overlap Scenarios

Overlap Levels: **: 1 view** **: 2 view** **: 3 view**

How about videos?

SLAM
for sequences
designed for real-time
causal, limited robustness
fast and efficient
limited self-calibration

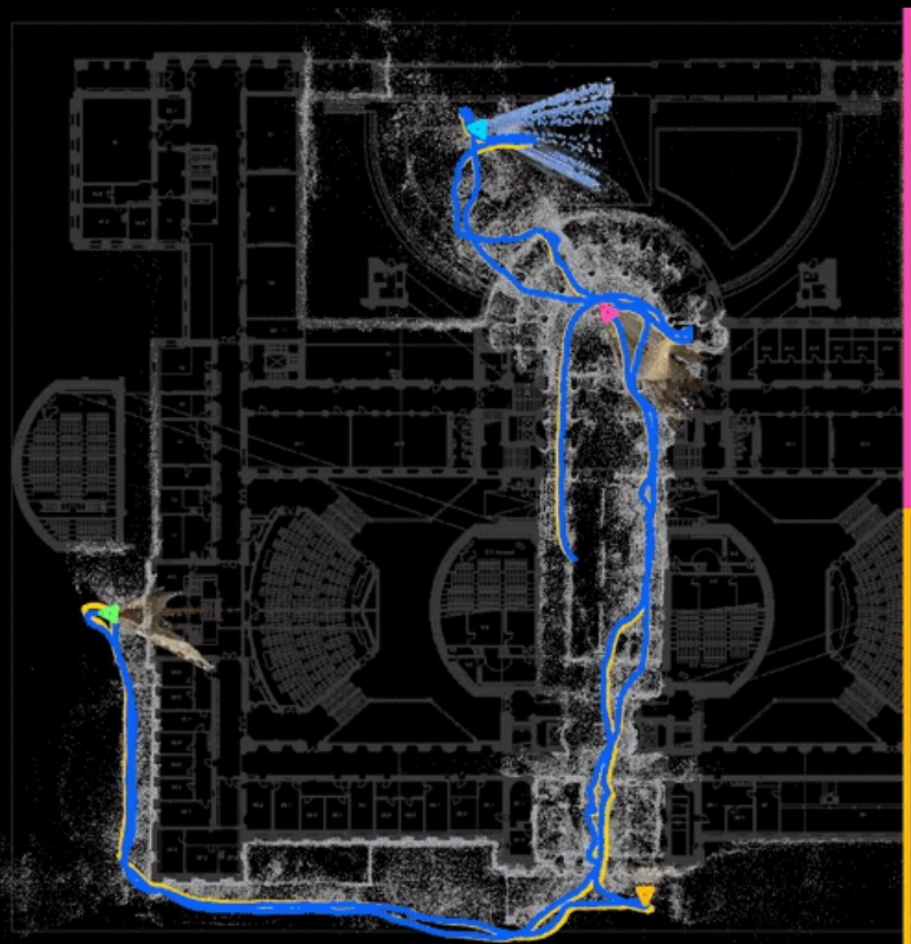
Structure-from-Motion
for unordered collections
designed for robust offline processing
treats videos like bags of images
slow for large inputs
can process uncalibrated inputs

Can't we have the best of both?

Our answer: VidMap [ECCV'26]

Zador Pataki

Marc Pollefeys



How it works

SLAM

❌ **incremental/causal**

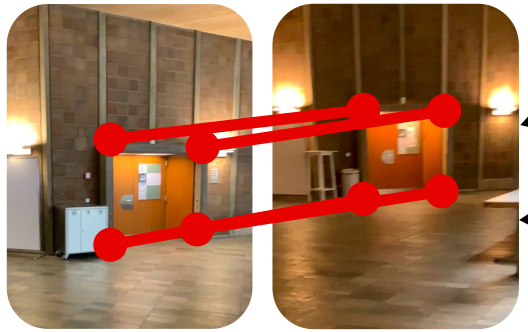
cannot recover from large drift

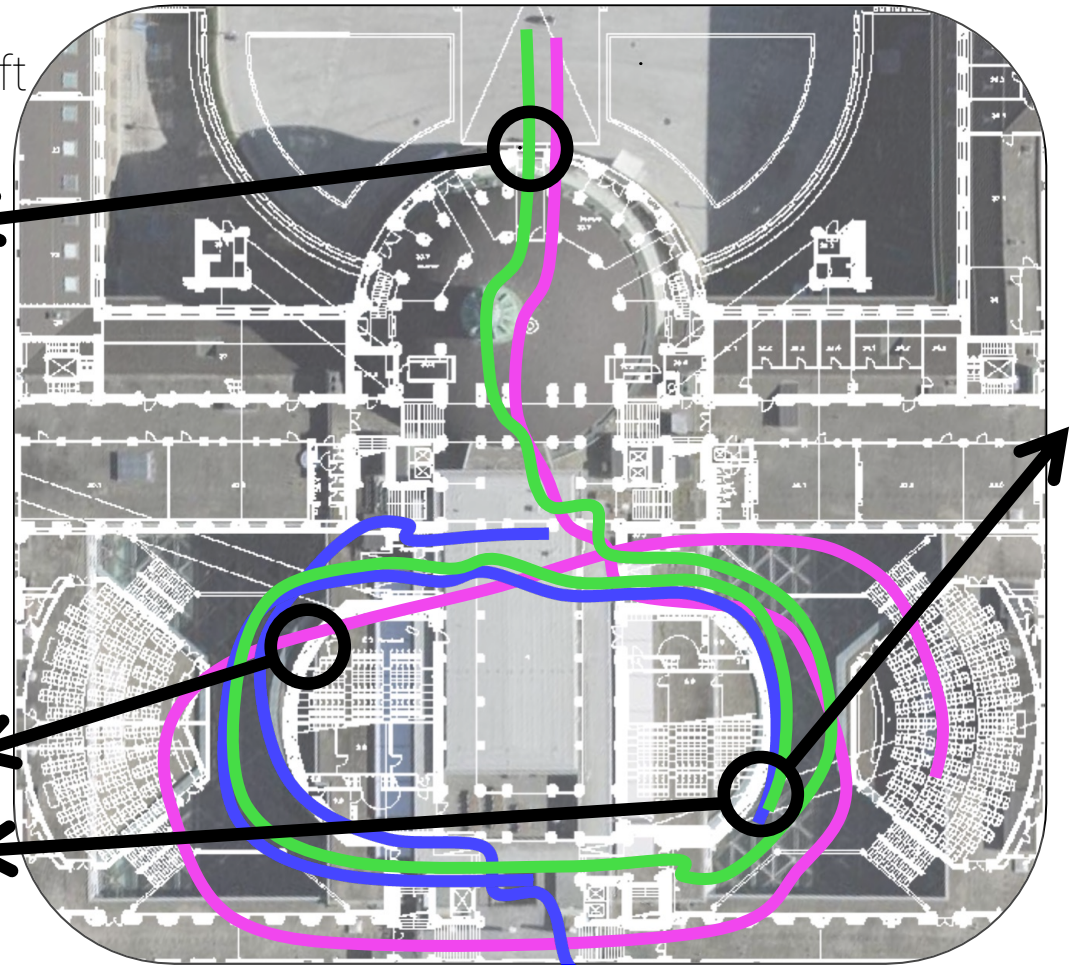
challenging motion ➡ drift

SfM

❌ **no temporal constraints**

symmetries ➡ failure





VidMap

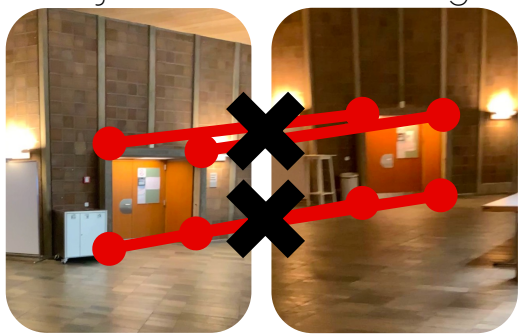
✅ **global/acausal**

loop closure upfront

➡ prevents drift

✅ **temporal constraints**

reject visual aliasing



How it works

SLAM

❌ **incremental/causal**

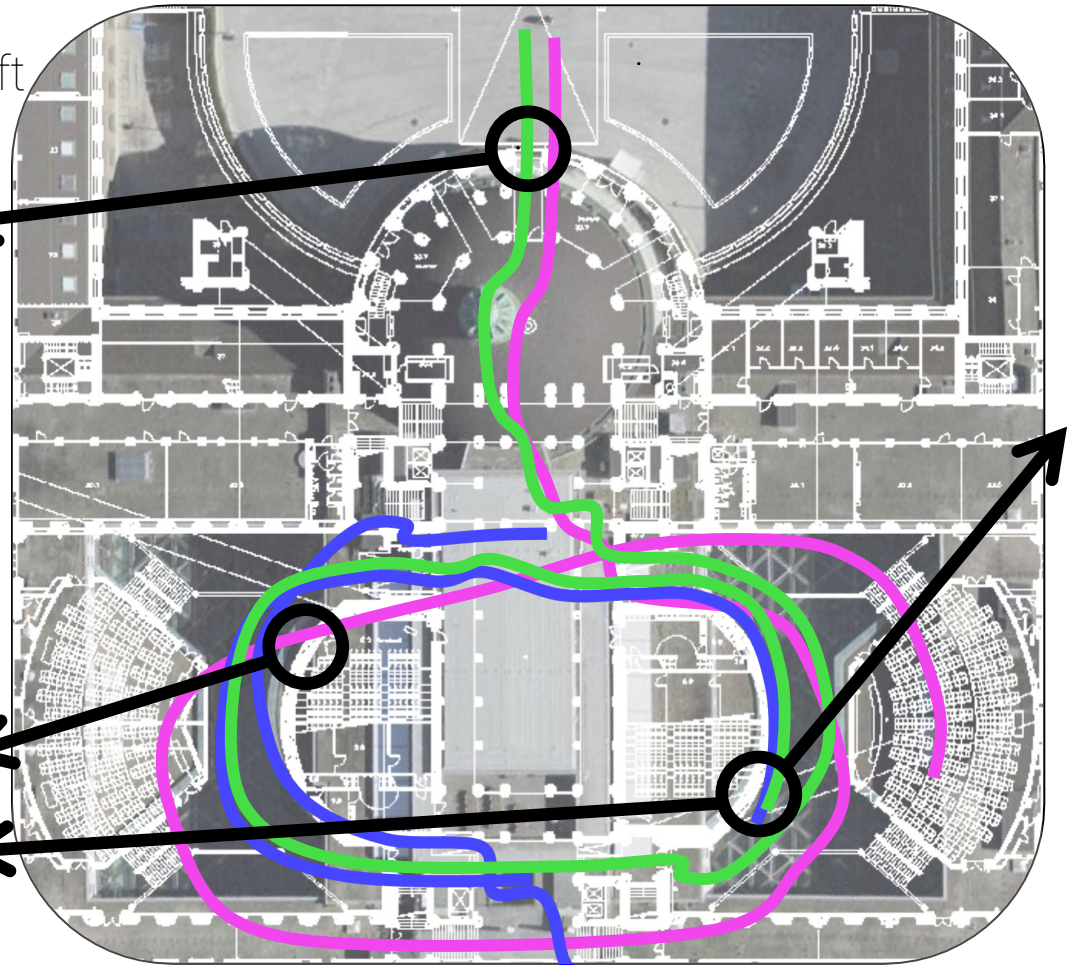
cannot recover from large drift

challenging motion ➡ drift

SfM

❌ **no temporal constraints**

symmetries ➡ failure



VidMap

✅ **global/acausal**

loop closure upfront

➡ prevents drift

✅ **temporal constraints**

reject visual aliasing

How it works

SLAM

❌ **incremental/causal**

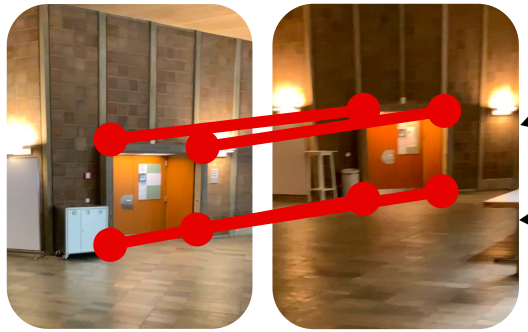
cannot recover from large drift

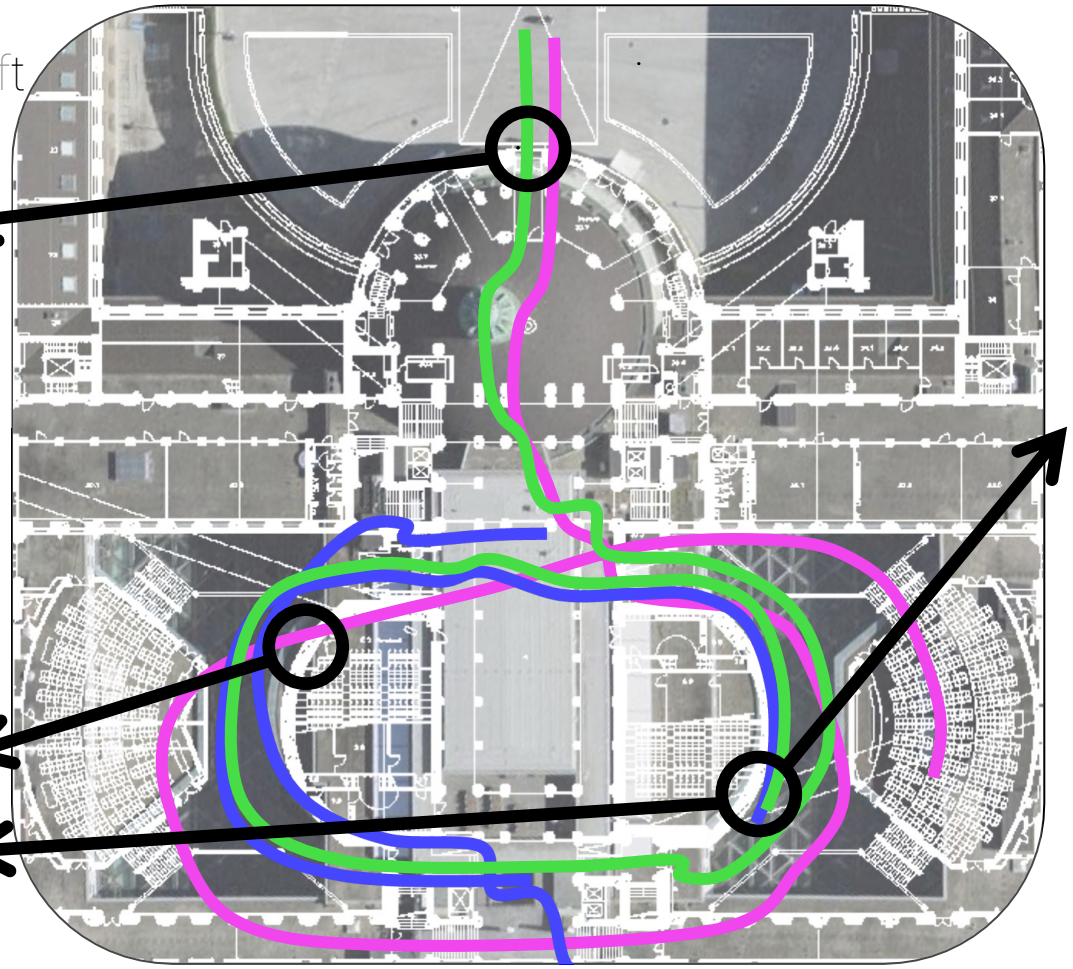
challenging motion ➡ drift

SfM

❌ **no temporal constraints**

symmetries ➡ failure





VidMap

✅ **global/acausal**

loop closure upfront

➡ prevents drift

✅ **temporal constraints**

reject visual aliasing

How it works

SLAM

❌ incremental/causal

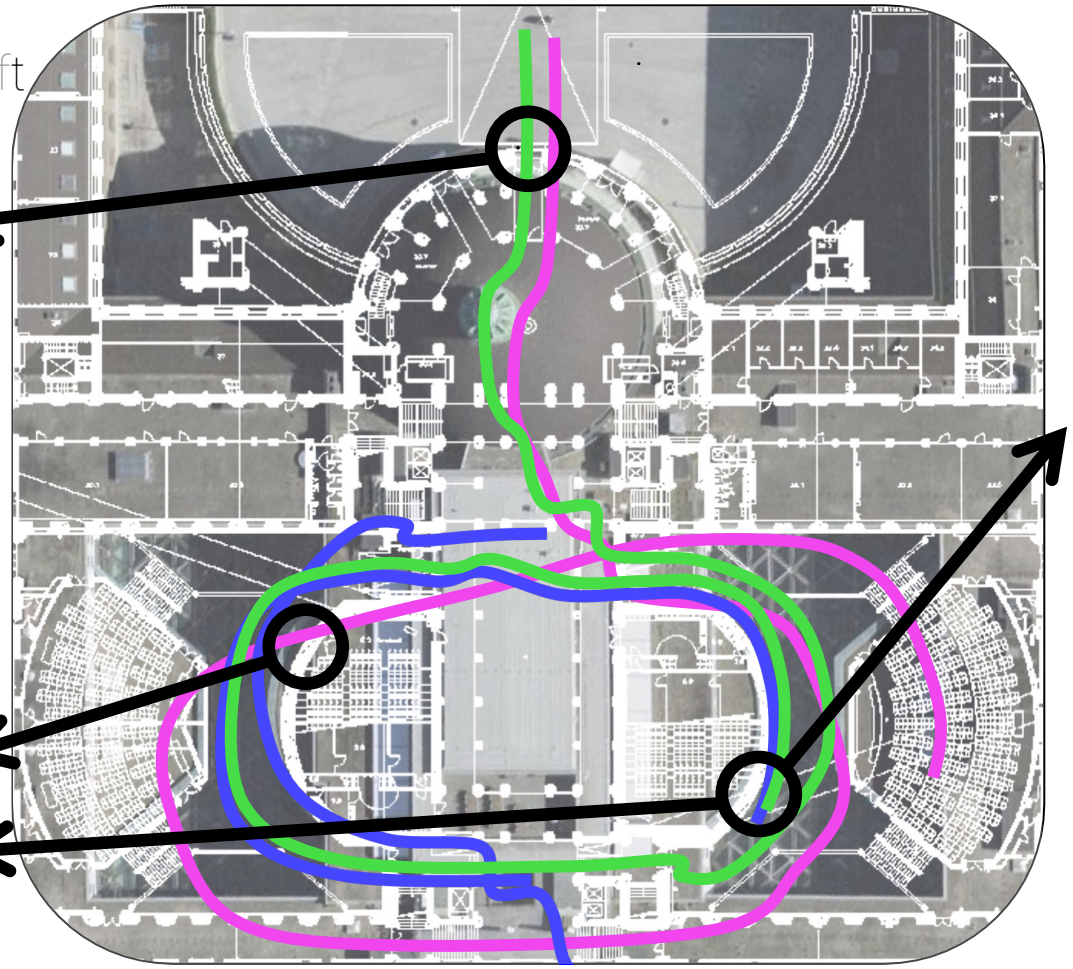
cannot recover from large drift

challenging motion → drift

SfM

❌ no temporal constraints

symmetries → failure



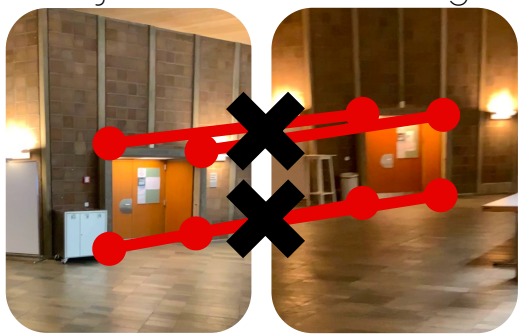
VidMap

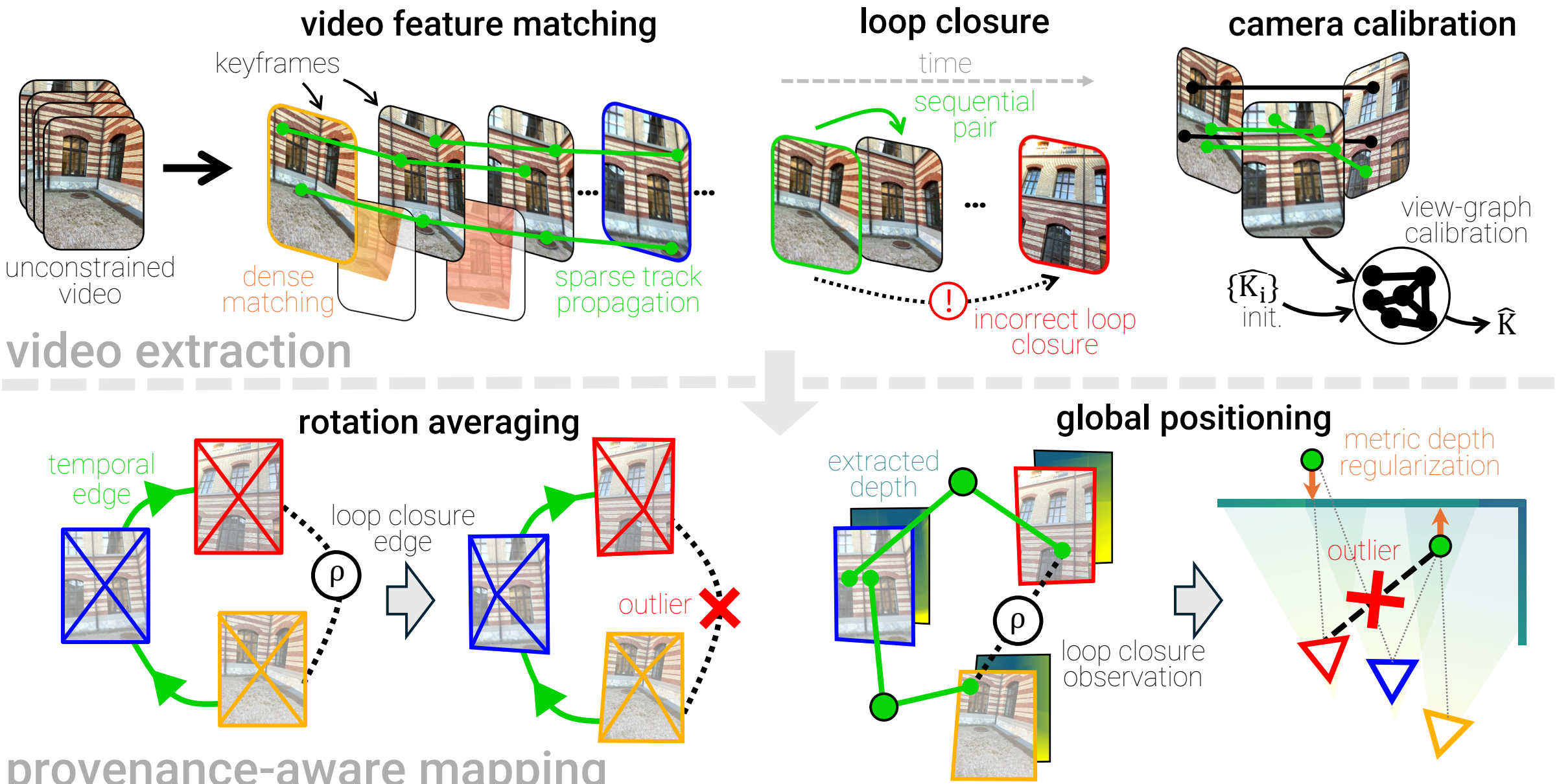
✅ global/acausal

loop closure upfront → prevents drift

✅ temporal constraints

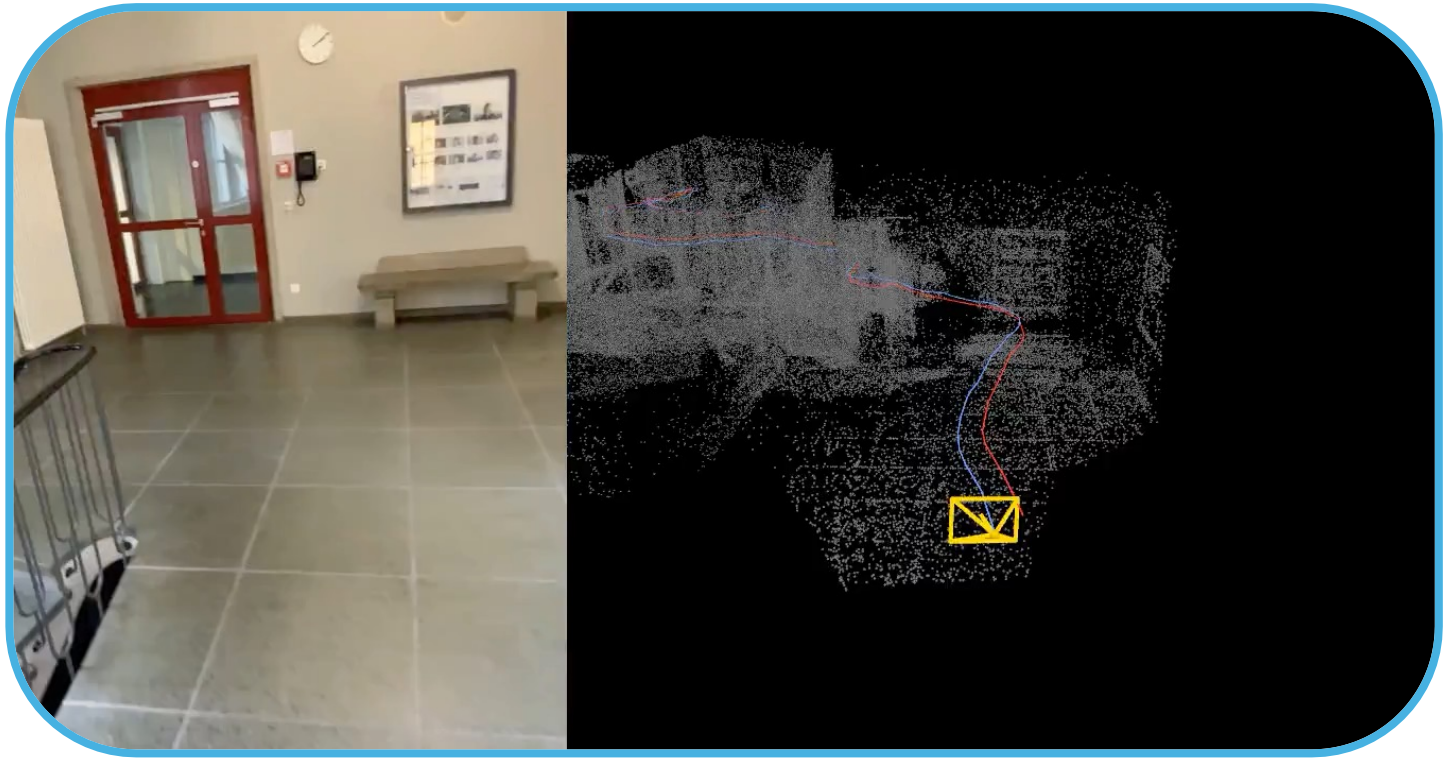
reject visual aliasing




video feature matching
keyframes
unconstrained video
dense matching
sparse track propagation
loop closure
time
sequential pair
incorrect loop closure
camera calibration
view-graph calibration
{K̂ᵢ} init.
K̂
video extraction
rotation averaging
temporal edge
loop closure edge
outlier
global positioning
extracted depth
loop closure observation
metric depth regularization
outlier
provenance-aware mapping

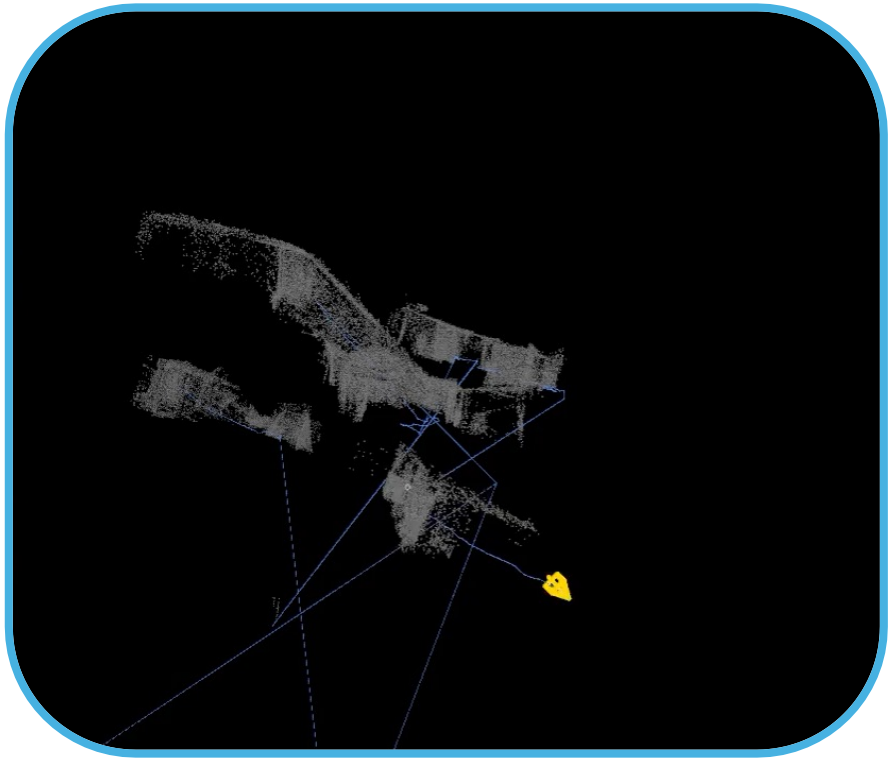
Addressing failures of classical approaches

Tracking loss:



tracks

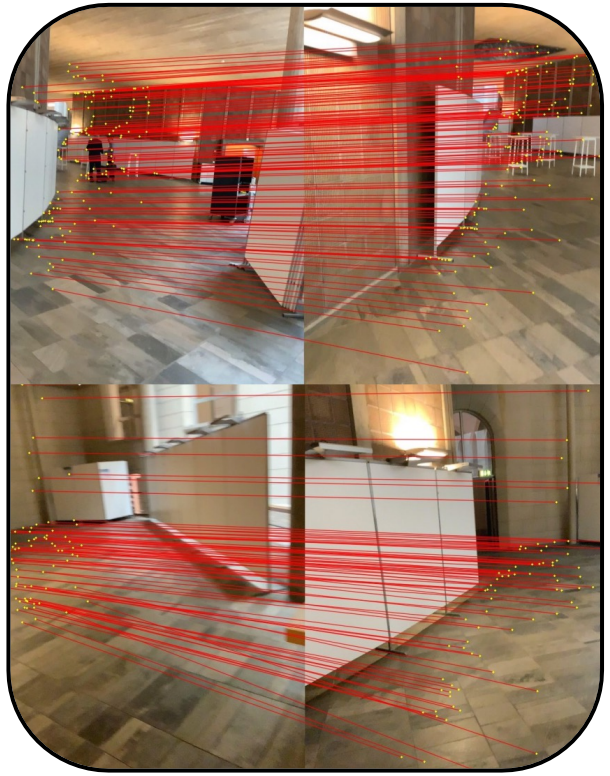
VidMap



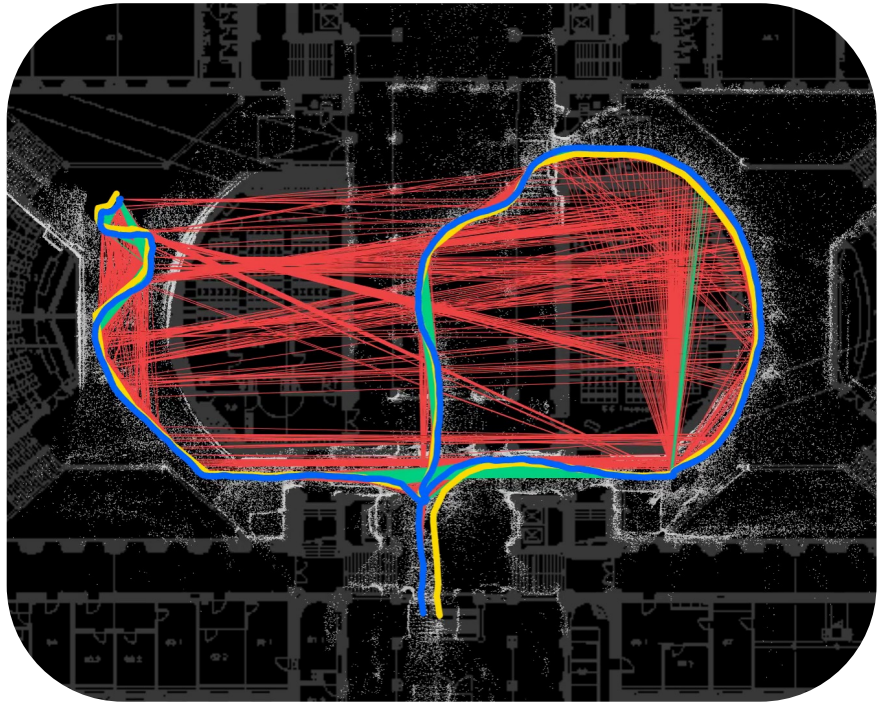
with sparse matching

Addressing failures of classical approaches

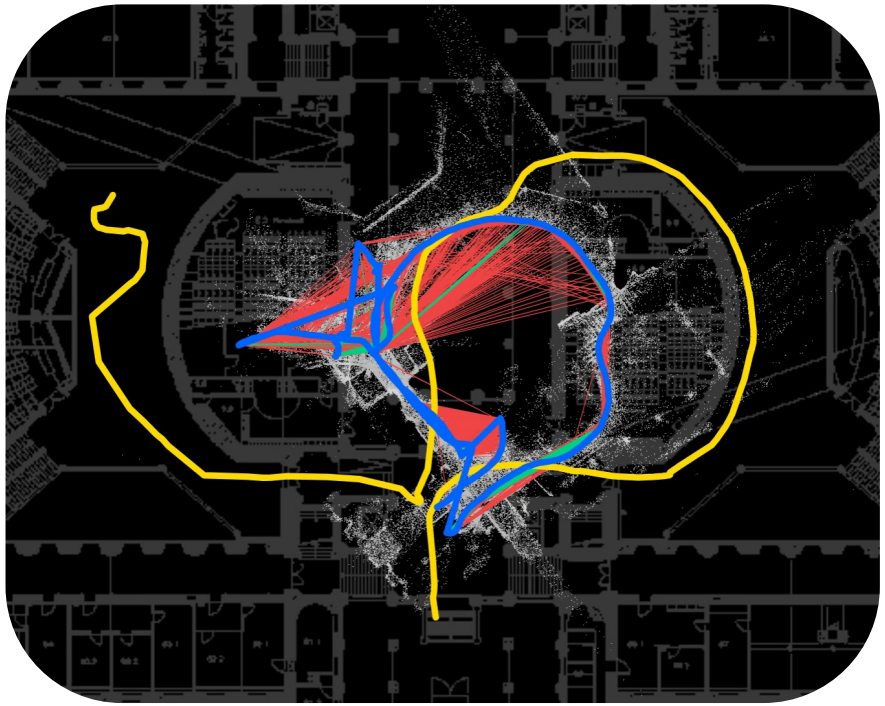
Symmetries – visual cues aren't sufficient!



visual aliasing



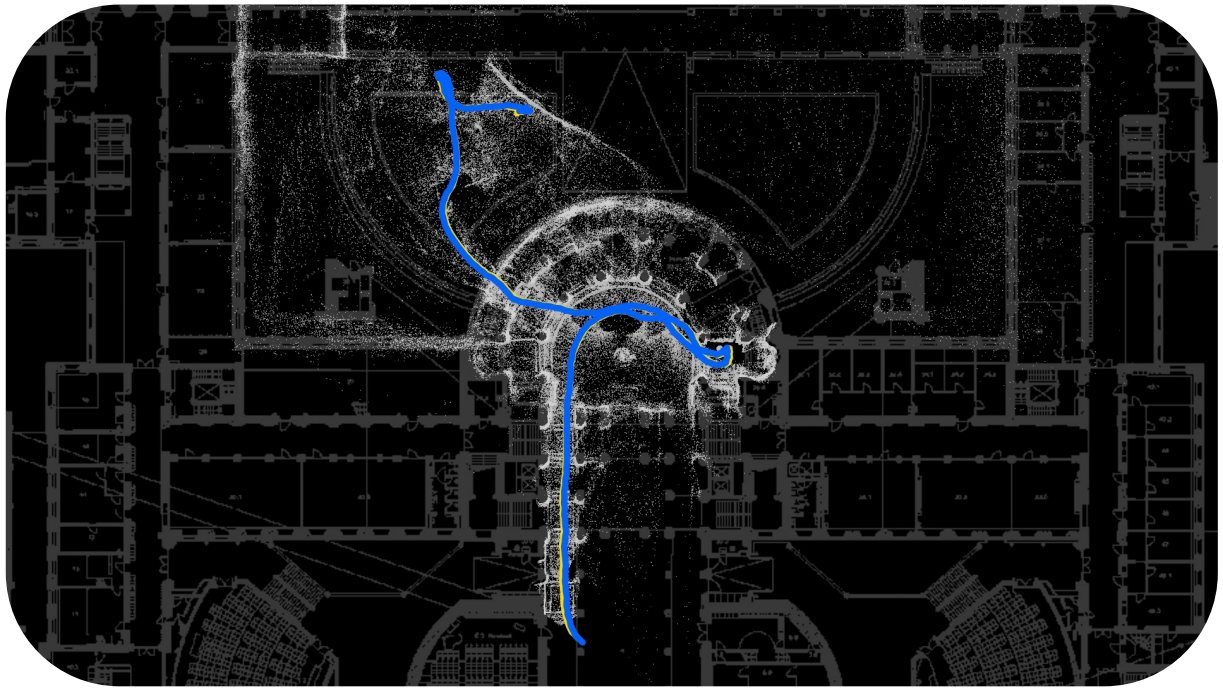
VidMap



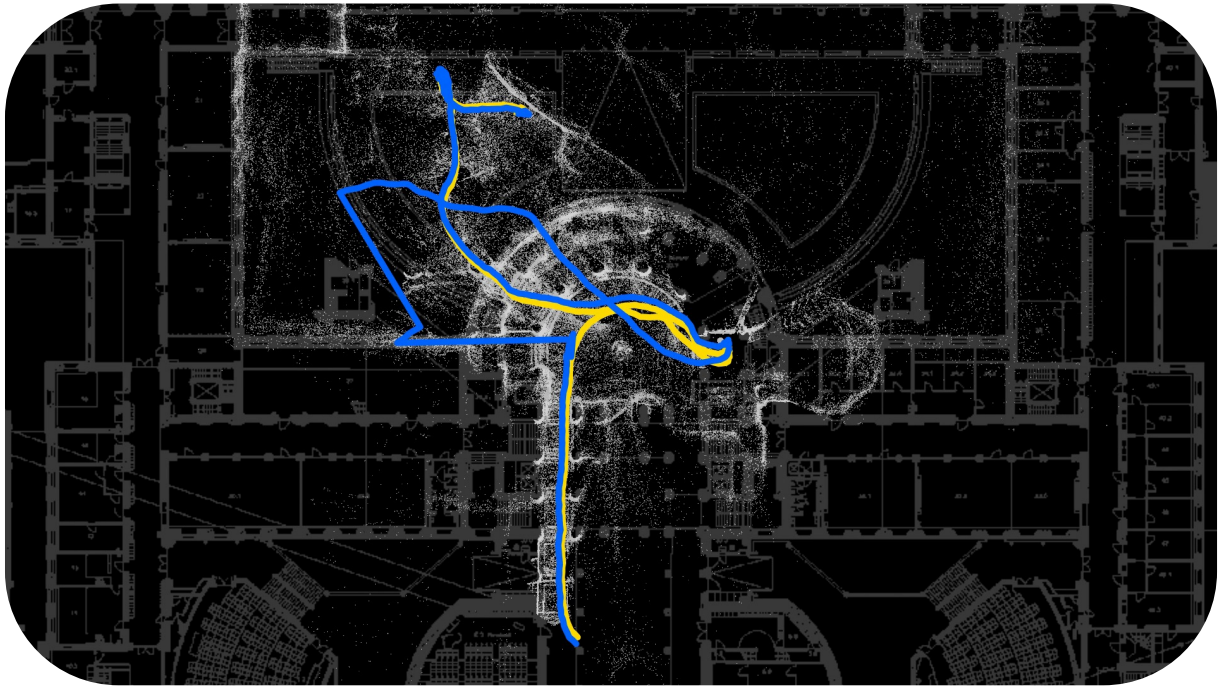
without
temporal provenance

Addressing failures of classical approaches

Degenerate motion & 3D scene:



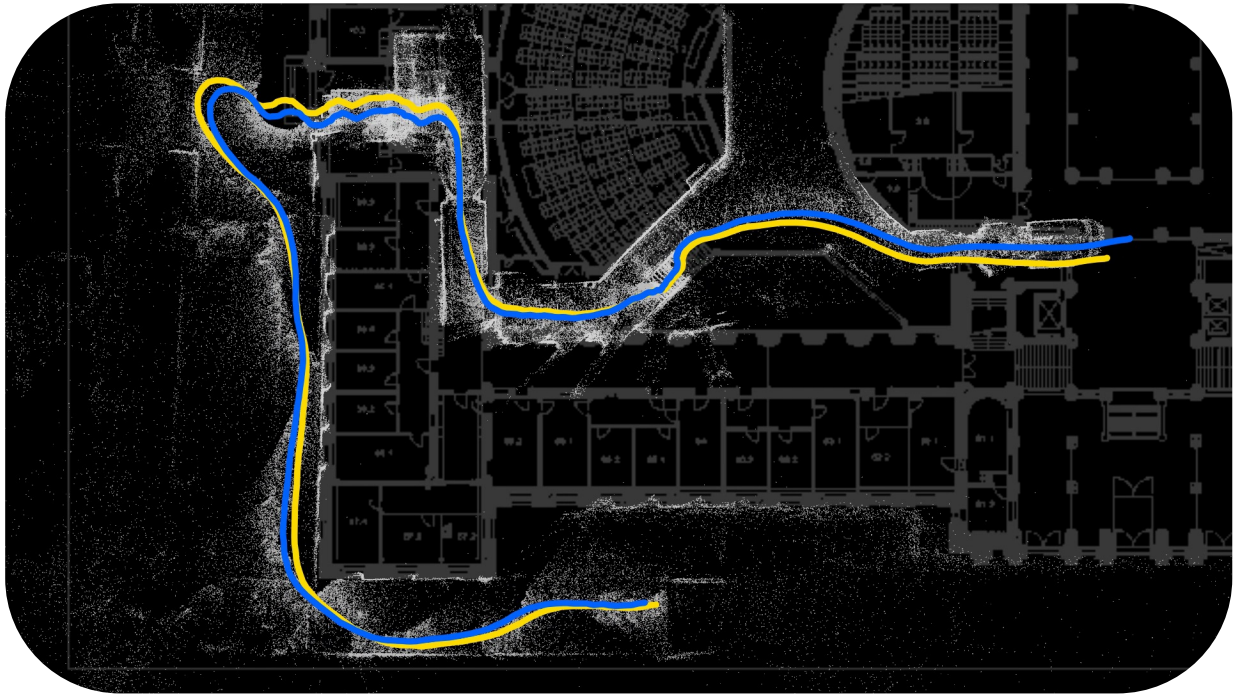
VidMap



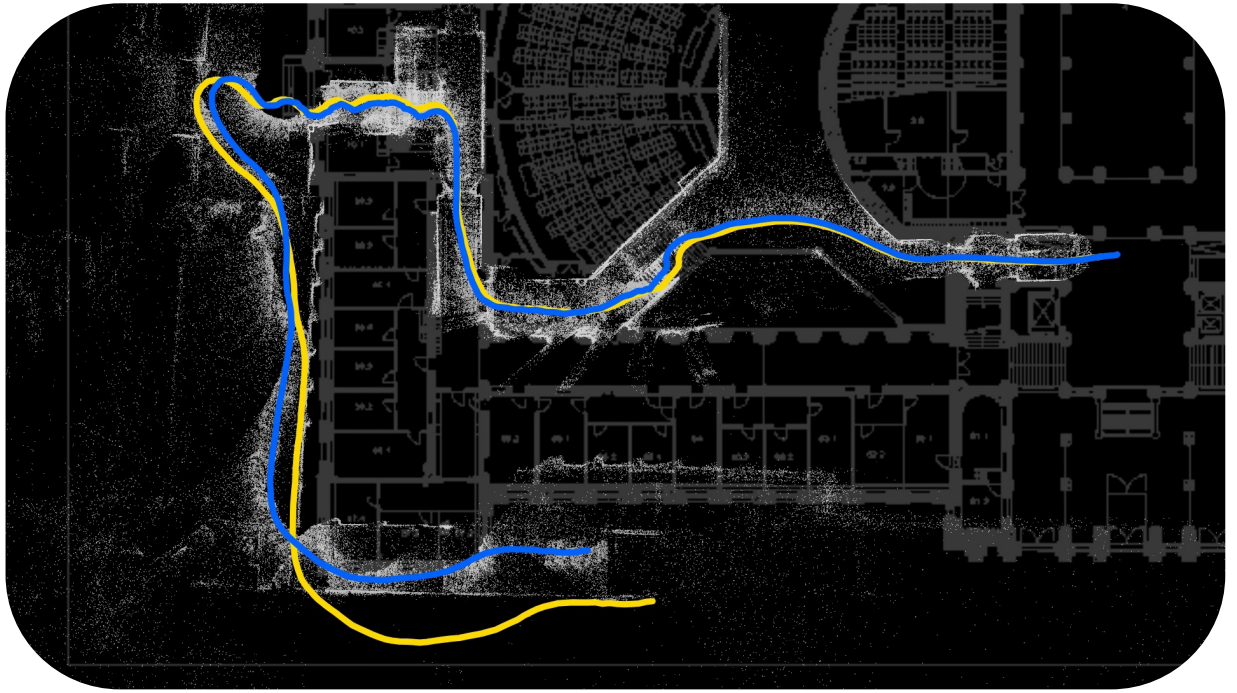
without depth regularization

Addressing failures of classical approaches

Scale drift:

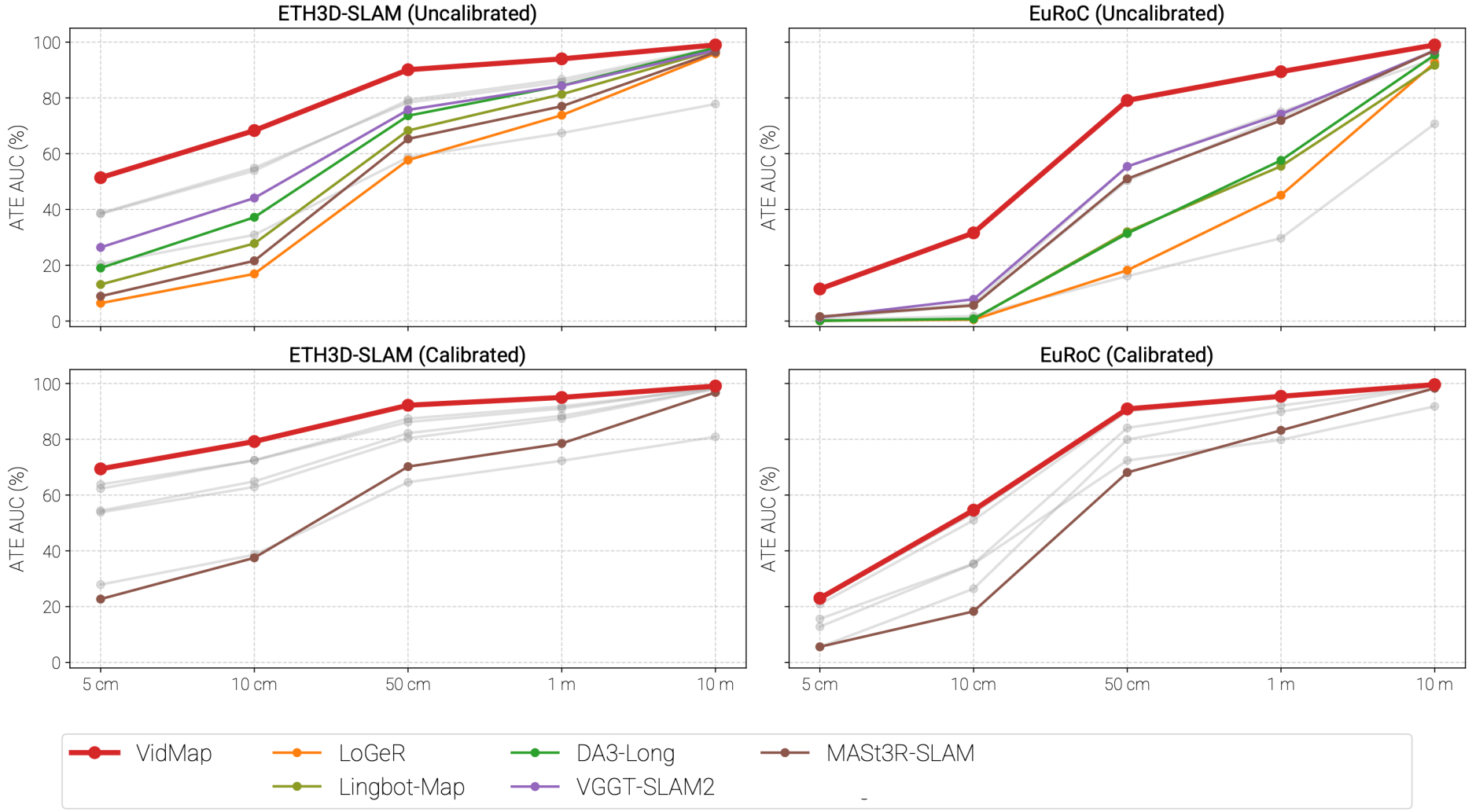


VidMap



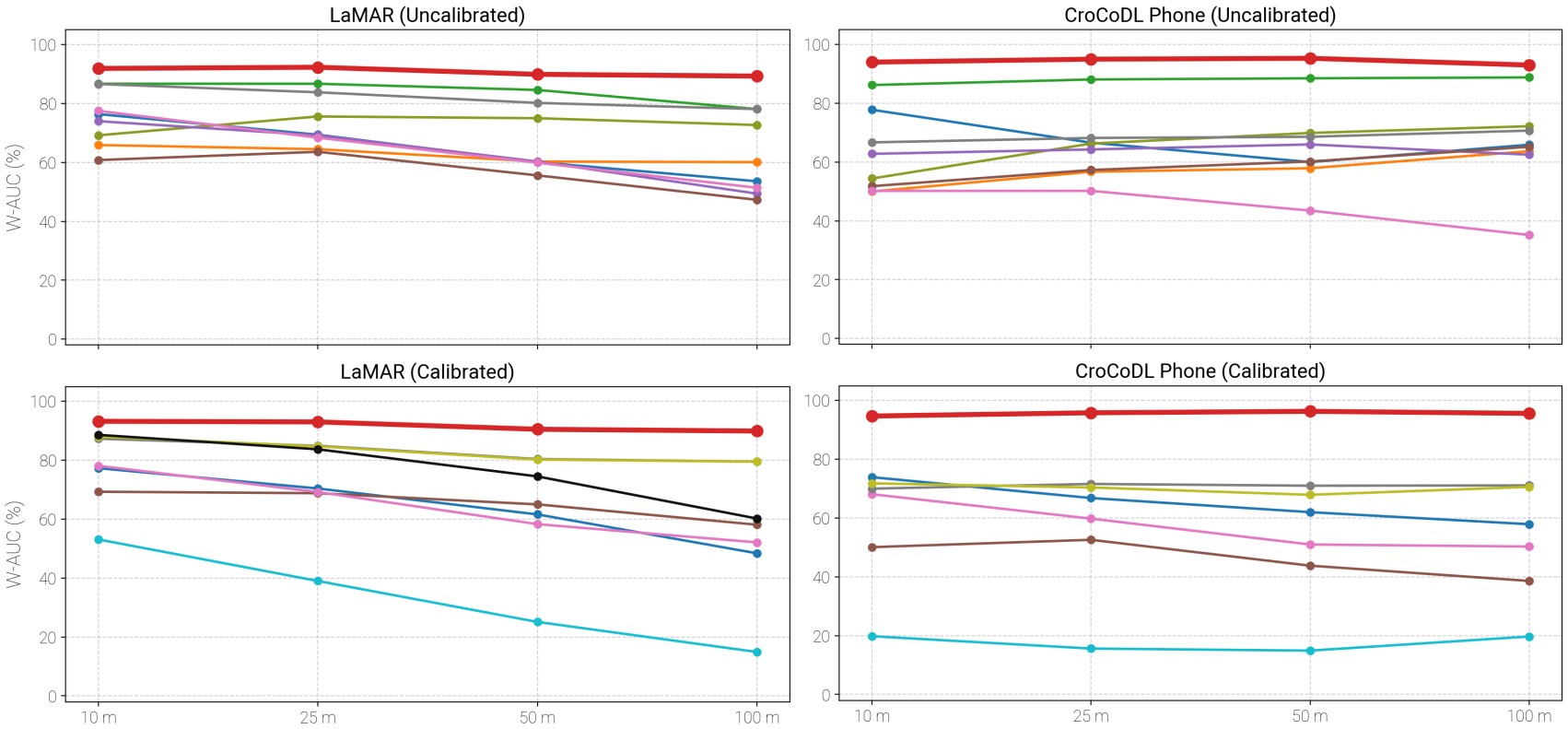
without depth metric scale

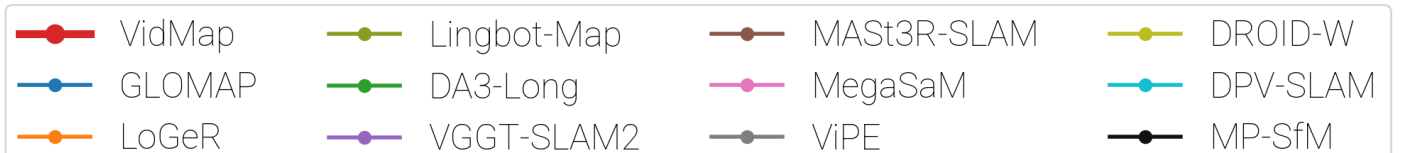
SLAM benchmarks are saturated



But E2E approaches are still behing in accuracy!

Harder benchmark push the limits





2 key factors:

image matching

Sparse: *LightGlue*

- Fast but brittle

Dense: *RoMa*

- Robust & subpixel accurate
- Unnecessarily slow

Old problems are saturated, right?

symmetry disambiguation

Geometry: *robust costs functions*

- Fast but brittle

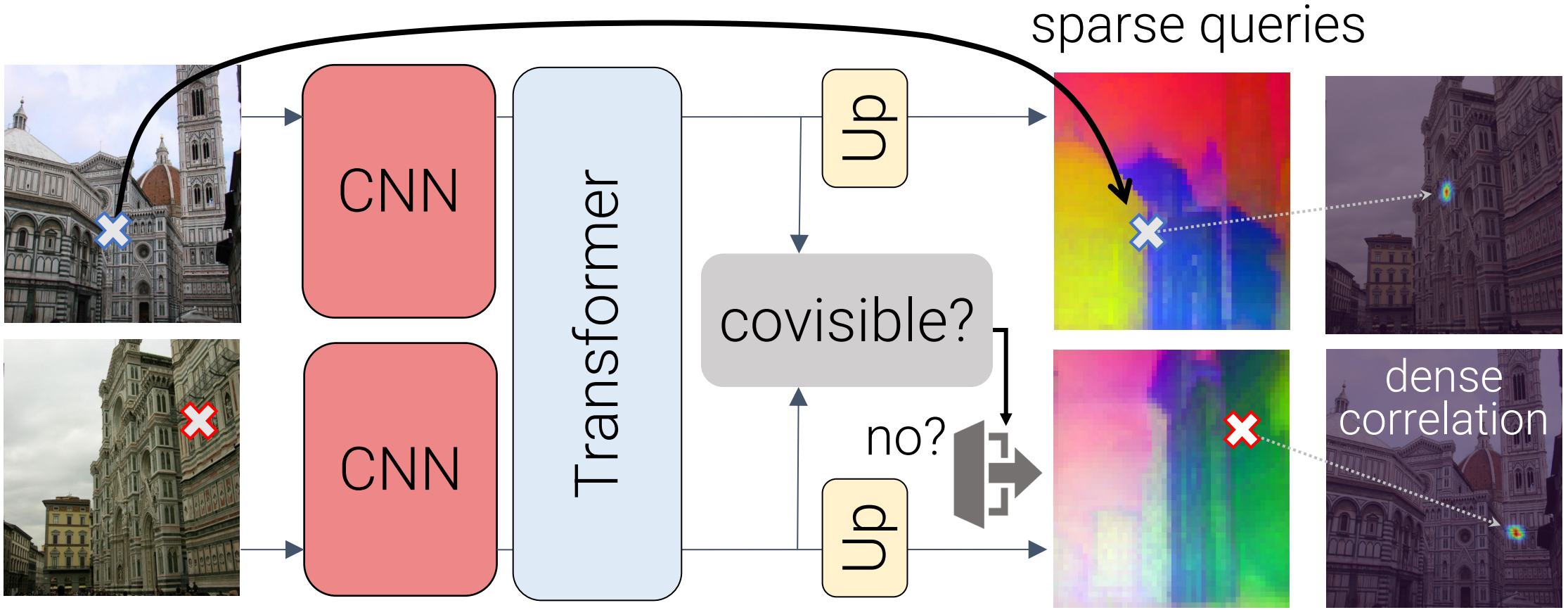
Matching: *inlier count*

- Free but brittle (recall >> precision)

Learned: *DoppelGanger++*

- Robust but slow

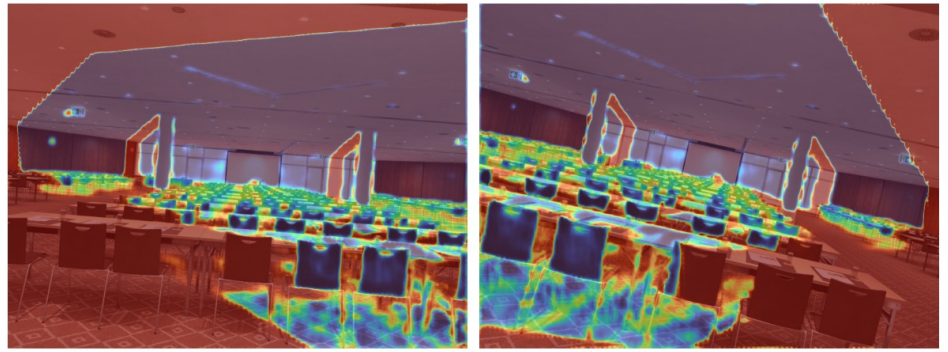
FlexGlue: Fast Matching & Disambiguation



- One model for sparse-to-dense, sparse-to-sparse, dense-to-dense
- Full image context helps both matching & disambiguation – no keypoints!

sparse-to-sparse

sparse-to-dense



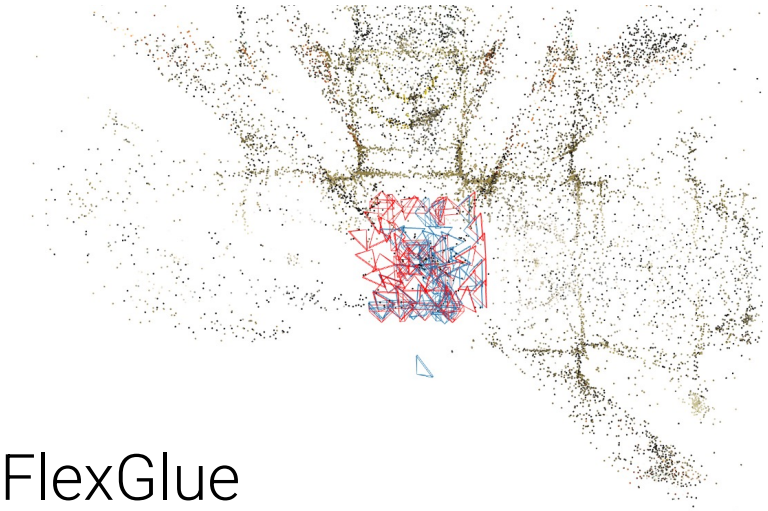
dense matchability heatmap

matches filtered

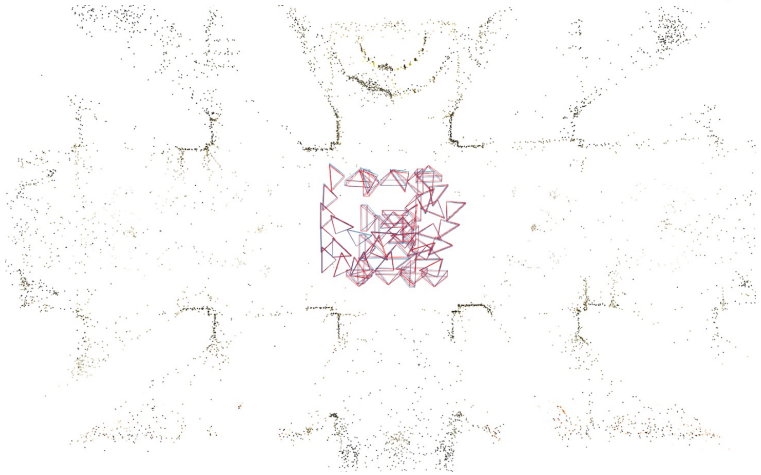
with Philipp Lindenberger

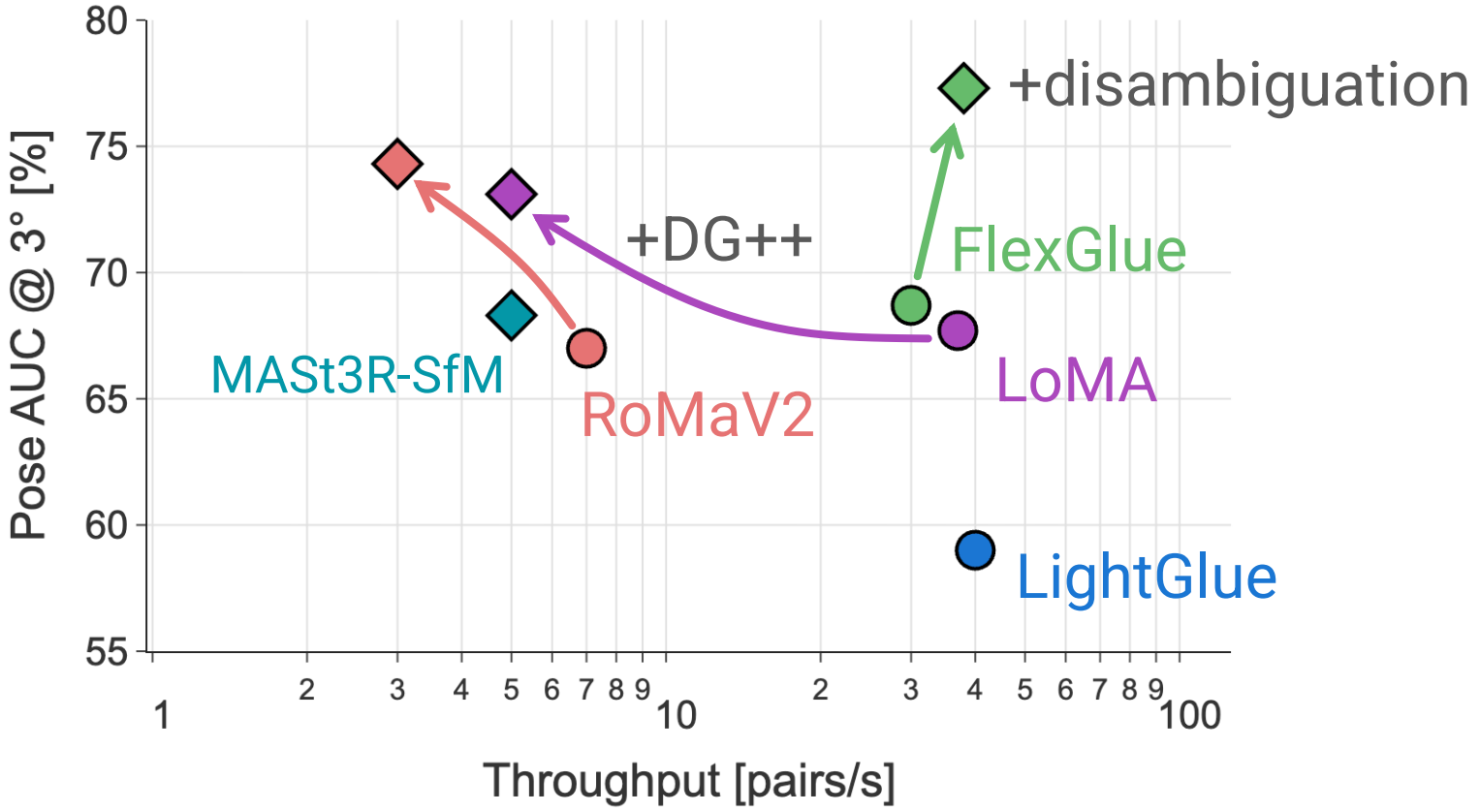
FlexGlue: designed for end applications

LoMA



FlexGlue





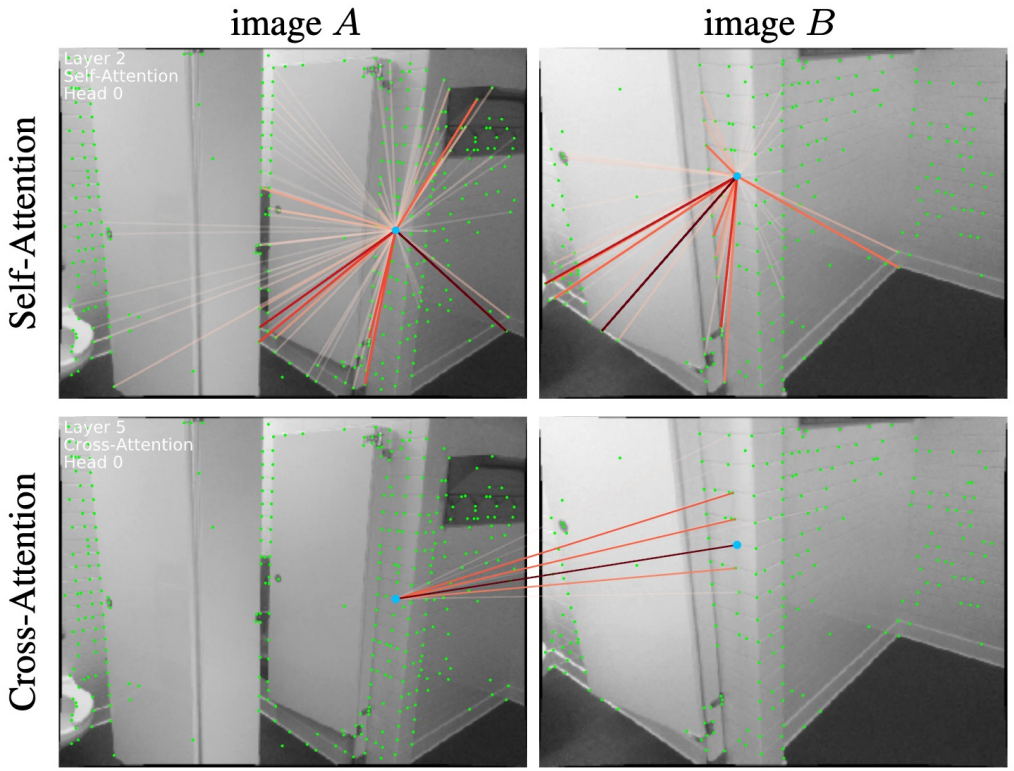
Disambiguation is critical in practice but conveniently ignored by all matchers.

Conclusion:
Better COLMAP = better data

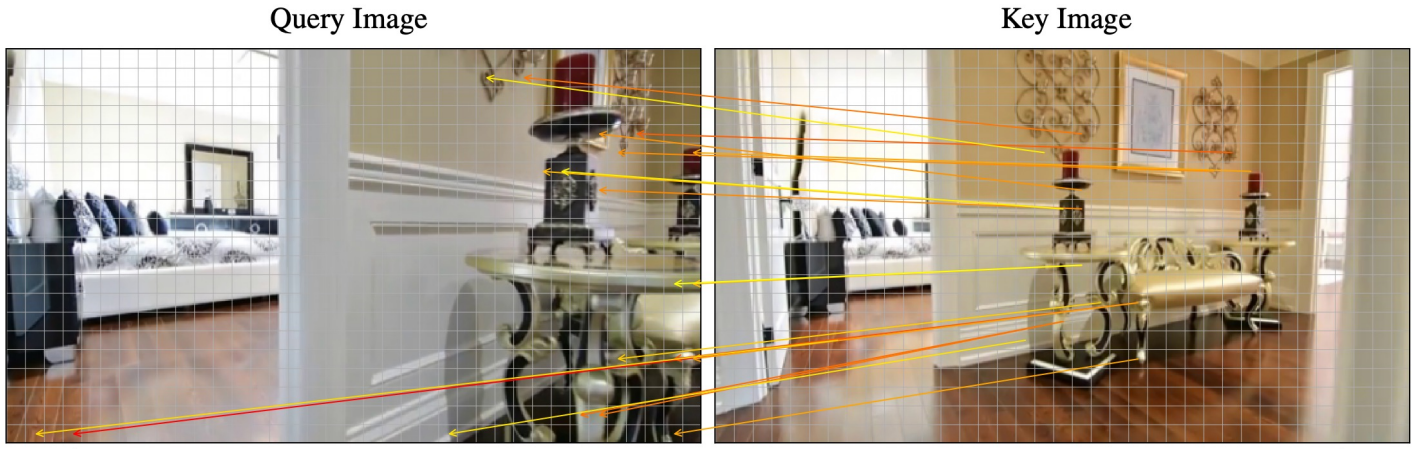
How about the model design?

Why is the precision of e2e models limited?

- Implicit correspondences & optimization with cross-attention
- Fast because low-resolution patches rather than interest points



2020: x-attention for matching



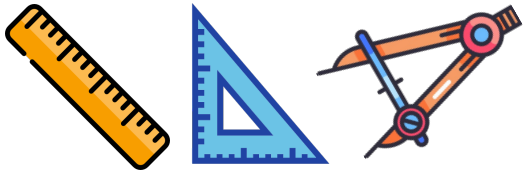
2025: implicit matching via x-attention

[Wang et al, CVPR 2026]

Why is the precision of e2e models limited?

- Precision is sacrificed to speed & throughput
  - But models are useful only past some accuracy levels!
- Geometric optimization is supposedly too slow
- We've already see this argument once…

3D geometry



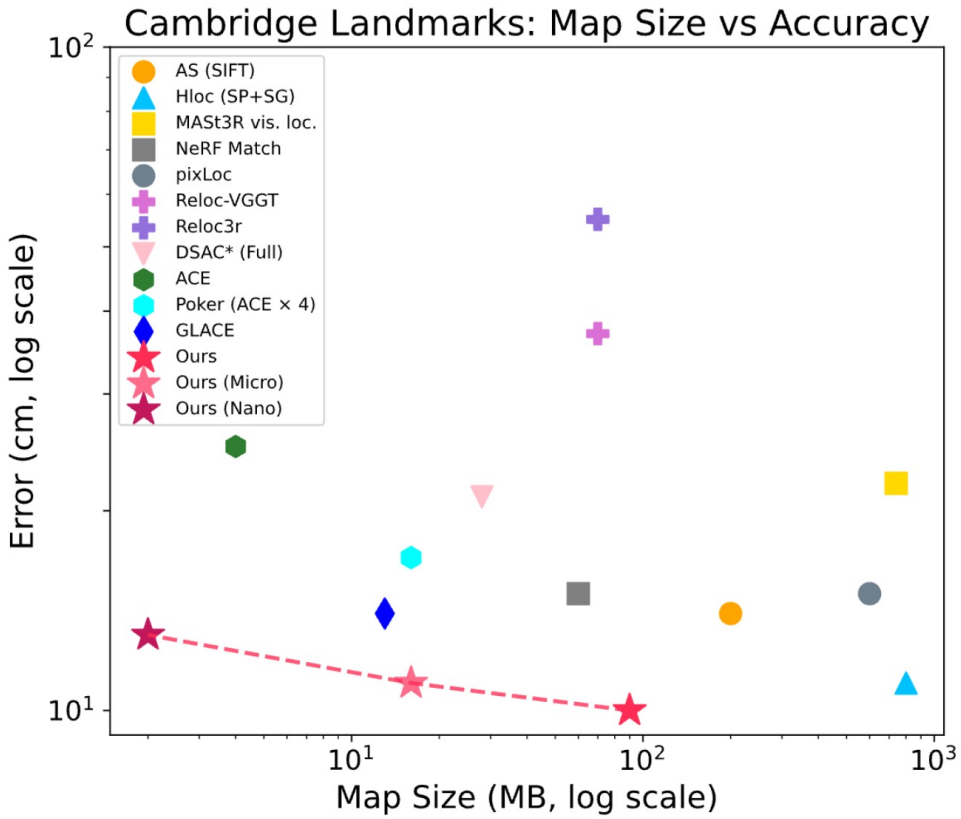
vs

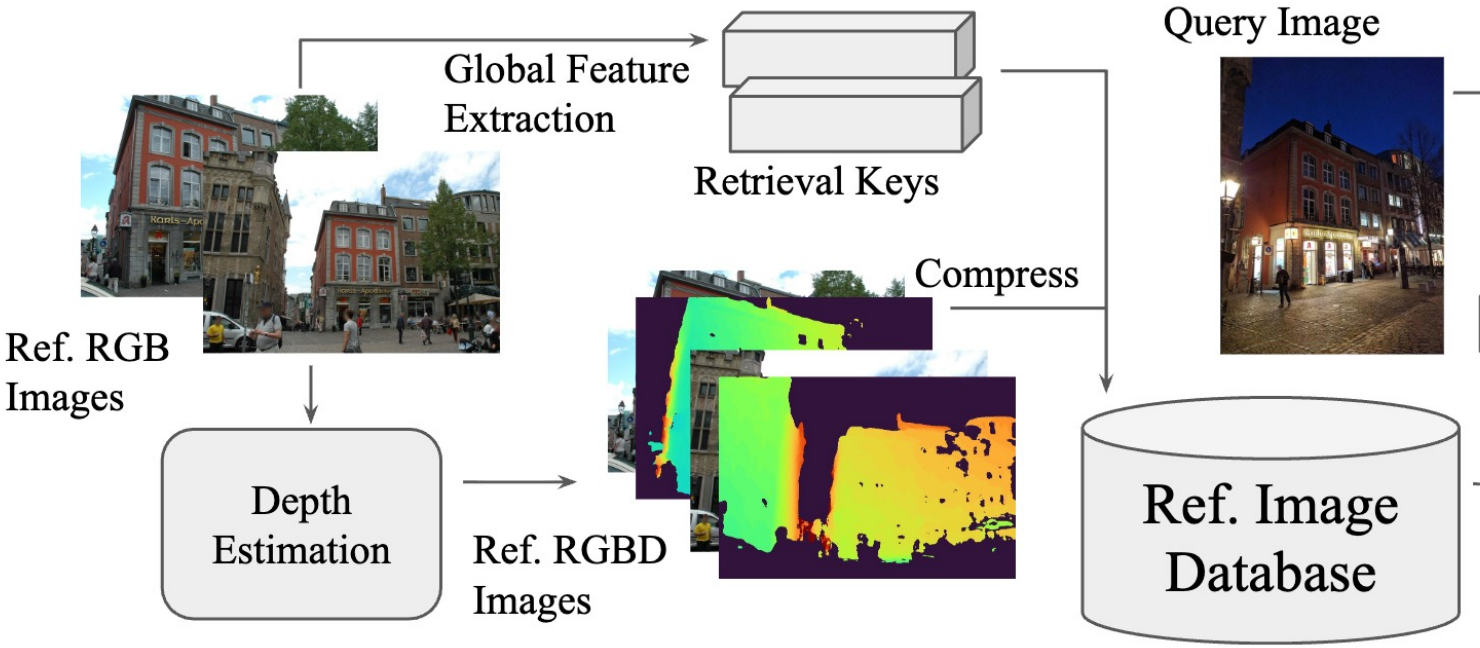
black box
deep learning

Scene representations in visual localization

- ImLoc: only store images and depth maps
- Can be compressed well, down to ~15KB/image
- DNNs are robust to compression artifacts

with Xudong Jiang

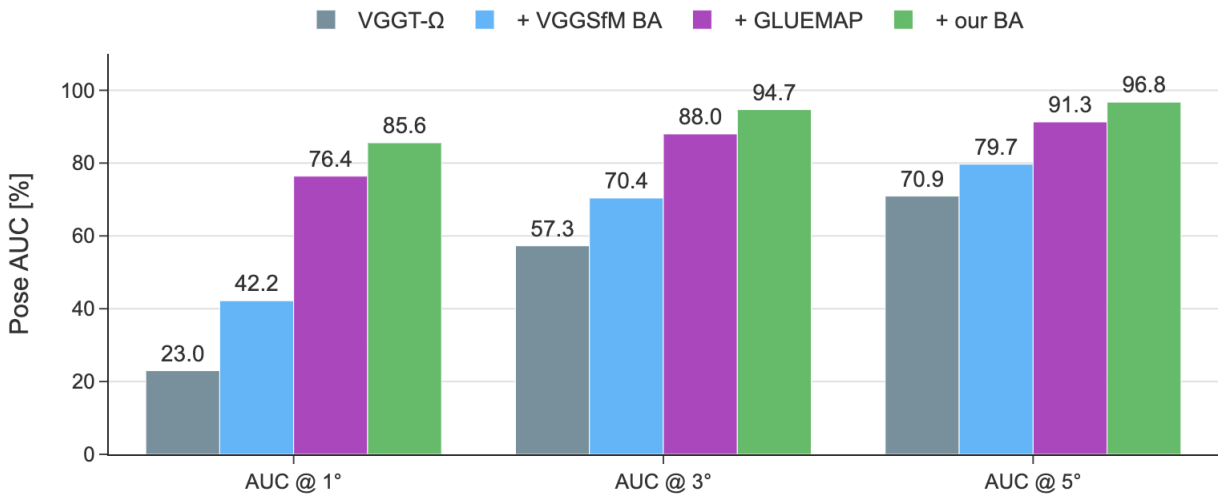


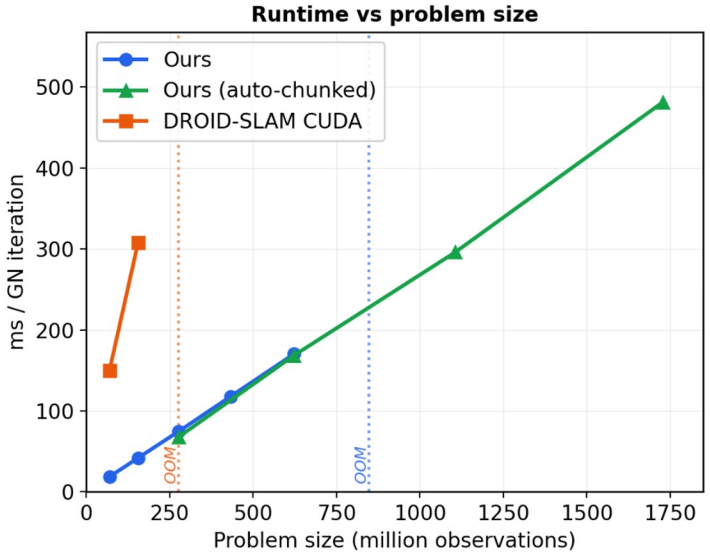


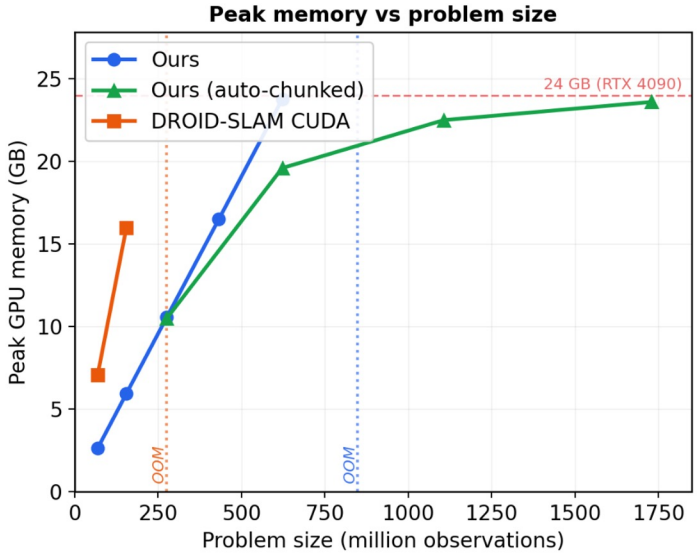
[Xudong Jiang et al., arXiv 2026]

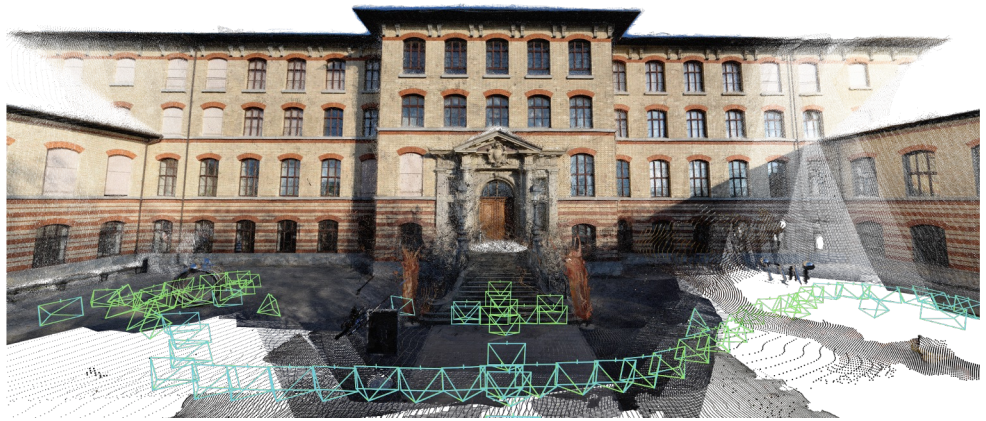
Bundle Adjustment can also be optimized

- Has haven't been as much tuned as PyTorch, especially for a GPU
- We reach linear time & constant memory, ~cost of e2e inference
- Boosts the accuracy of e2e models









230M observations, 70ms/iter

Let’s take a step back…

How are LLMs solving arithmetics?

How we solved arithmetics with LLMs

Fast calculators

2020

"Calculators are too slow/rigid, let's have LLMs do arithmetic"

"Oh no, hallucinations!"

2023

Tool use

LLM writes equation, calculator solves it

Feedback loop

LLM checks output & corrects mistakes

2024

2025

Agentic routing

LLM dynamically calls tools only when needed

Geometry isn’t different from arithmetics

- LLMs can use a tool rather than learning accurate calculations
  - They ***can*** learn it (scaling!), but it’s wasteful and much slower
  - Focus capacity on higher level abstraction, memorizing stuff
  - **The harness is the symbolic aspect of modern AI**

- LLMs and VGGT are both Transformers
  - Use optimization/geometry as a tool for accuracy
  - Let models focus their capacity on robustness & learned priors

Where is 3D reconstruction?

Fast calculators

2020

"Calculators are too slow/rigid, let's have LLMs do arithmetic"

"Oh no, hallucinations!"

Tool use (BA)

Feedback loop

Agentic routing

We are here!

What will take us further?

- A better **harness**: maybe several thousands of LoC (Claude Code)
- Helping Transformers understand the optimization process
- Unlike LLMs, inputs & outputs live in different spaces: images vs 3D

Constraints:

- don't make the training overly expensive
- don't bottleneck information or impair convergence
- don't introduce failures when geometric models don't fit the data

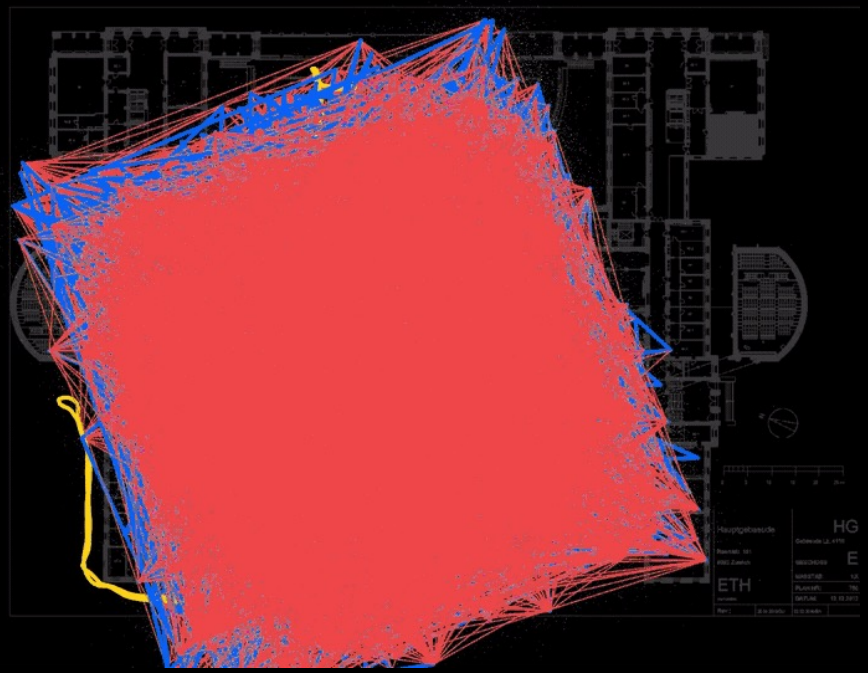
Conclusions

- Obsolescence vs. Evolution
  - Classical geometry is behind the data and maybe soon the model too
- Hybrid Optimization Pipelines
  - We can do explicit optimization of learned priors
  - Unclear how to do this in a DNN
- Implicit vs. Explicit maps
  - Neural nets are good at compression, but so is JPEG XL

Thank you!

psarlin.com

VidMap: Poster #157, session #3, Friday AM
github.com/cvg/vidmap



Credits:
Zador Pataki
Philipp Lindenberger
Xudong Jiang