(Full) Attention is More Than You Need

Sparse Cross-View Attention in VGGT

Maria Pilligua

EPFL CVLab · Supervised by Aoxiang Fan & Pascal Fua

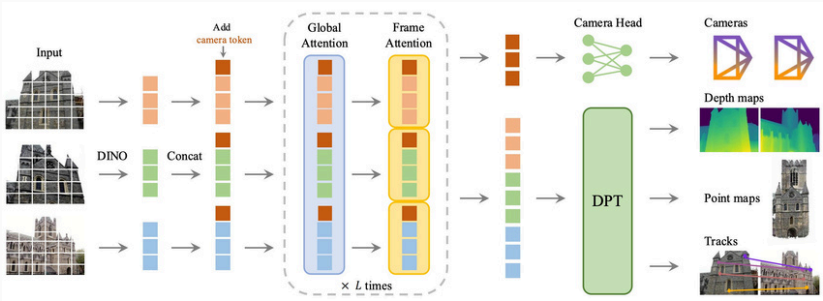
Outline

1

Motivation

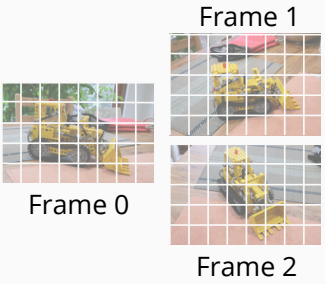
3D reconstruction is now transformer-based

Recent methods like VGGT recover cameras and dense geometry directly ftom many images.
They rely on **global** attention between *every* patch of *every* image



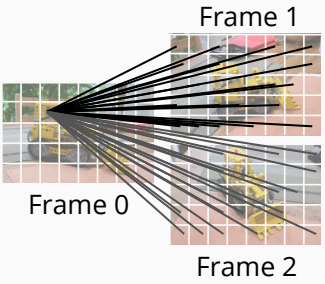
Global attention is a bottleneck

- **frame** attention: within an image (cheap)
- **global**: every patch attends to every patch of every image



Global attention is a bottleneck

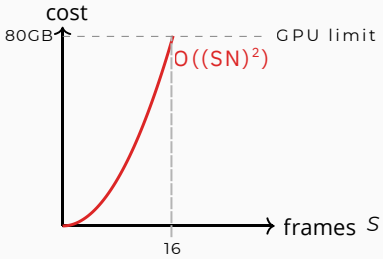
- **frame** attention: within an image (cheap)
- **global**: every patch attends to every patch of every image



cost grows $O((SN)^2)$

S: # patches per image

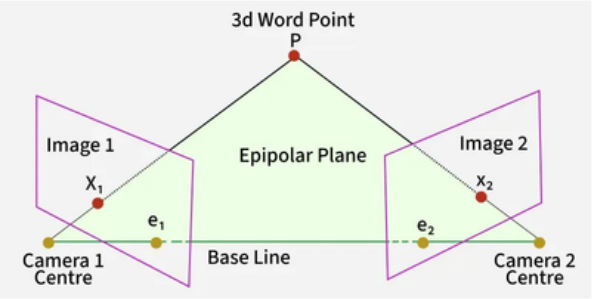
N: # images



But do we need every interaction?

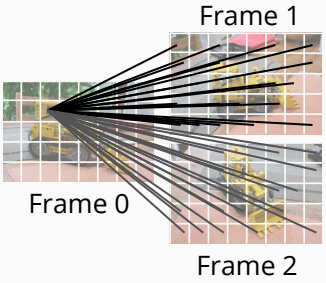
In classical geometry, camera poses are recovered entirely from **feature correspondences.**

So why should a transformer need every pairwise interaction?

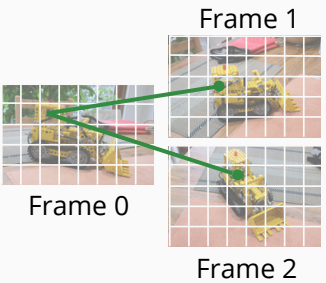


Idea: Sparse Global Attention

Full global attention

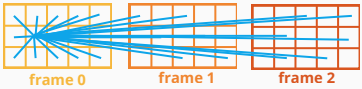


Sparse global attention

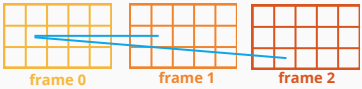


Attention should follow correspondences

Full global attention



Sparse global attention



Does this idea work?

Sparse attention can even outperform dense

trained and tested with **GT correspondances** (oracle)

The catch is circular

To know **where** to attend, we first need the correspondences.

But computing them is the exact $O((SN)^2)$ cost we are trying to remove.

The catch is circular

To know **where** to attend, we first need the correspondences.

But computing them is the exact $O((SN)^2)$ cost we are trying to remove.

Can we **learn** correspondences good enough for sparse attention, without paying the quadratic cost?

The catch is circular

To know **where** to attend, we first need the correspondences.

But computing them is the exact $O((SN)^2)$ cost we are trying to remove.

Can we **learn** correspondences good enough for sparse attention, without paying the quadratic cost?

Q1 how accurate must they be? · **Q2** where do they come from?

2

Methodology

Motivation > Methodology > Results > Conclusions

Q1: How accurate must correspondences be?

Q1: How accurate must correspondences be?

Perturbations:

- ***drop***: remove a true key
- ***swap***: repoints a key to a random token
- ***jitter***: moves a key by one patch

Q1: How accurate must correspondences be?

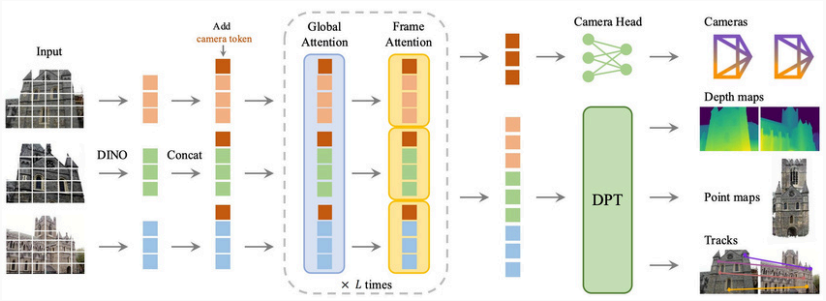
Perturbations:

- ***drop***: remove a true key
- ***swap***: repoints a key to a random token
- ***jitter***: moves a key by one patch

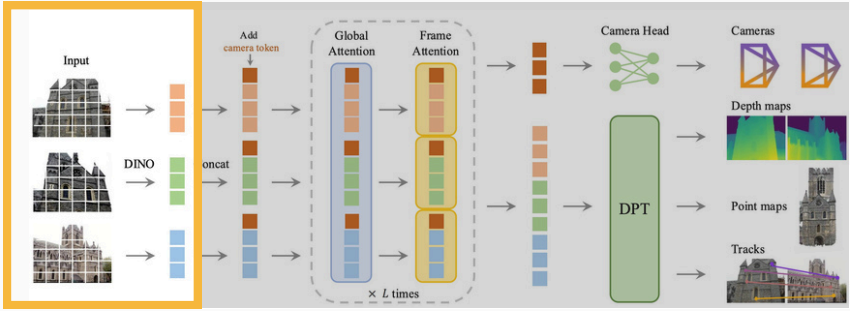
% index intact	pose AUC@30
100% (oracle)	**59.1**
75%	52.4
50%	50.1
VGGT baseline	50.2

Q2: Where do the matches come from?

Q2: Where do the matches come from?

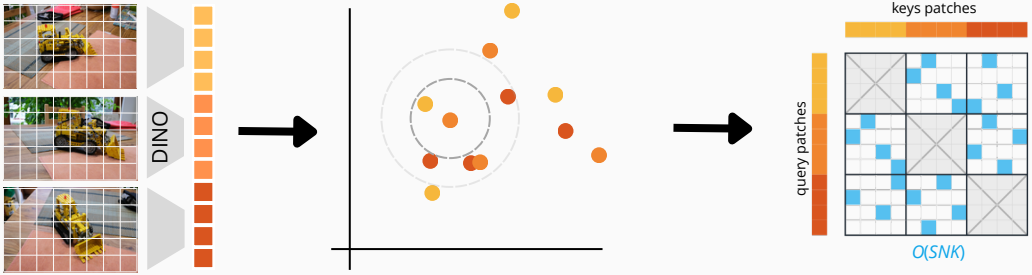


Q2: Where do the matches come from?



Q2: Where do the matches come from?

DINO features + **mutual NN**



Q2: Where do the matches come from?

DINO features + **mutual NN**

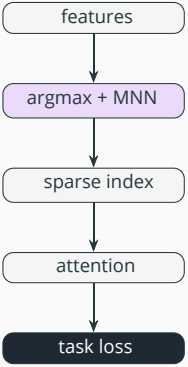
	precision	recall
MNN	41.1	27.3

mediocre (41% prec.)

The key insight: the gradient is blocked

we cannot finetune end-to-end

argmax is non-differentiable



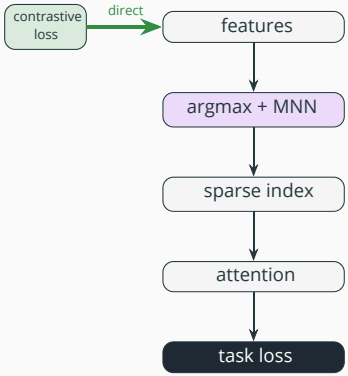
Finetuning Dino

we cannot finetune end-to-end

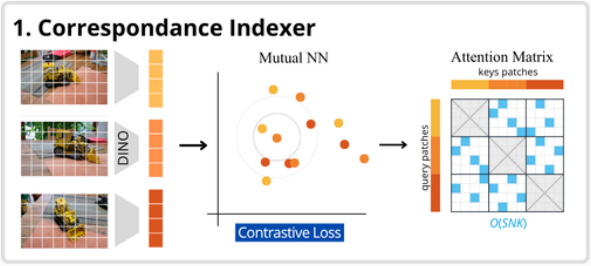
argmax is non-differentiable

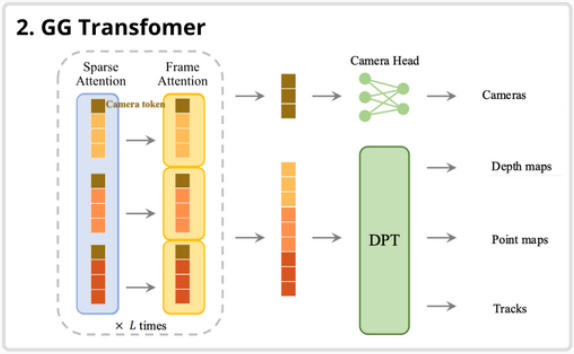
Fix

contrastive loss on features
(train-time only)



The full method





3

Results

Motivation > Methodology > Results > Conclusions

VGGT sparse beats dense, with no geometry at test time

	geometry at test?	pose AUC@30 ↑	median R ↓
VGGT (full attention)	–	50.2	2.6
VGGT sparse (oracle)	yes	59.1	2.1
VGGT sparse (frozen DINO)	no	38.5	3.4
VGGT sparse (FT DINO)	**no**	**55.1**	**2.0**

VGGT sparse beats dense, with no geometry at test time

	geometry at test?	pose AUC@30 ↑	median R ↓
VGGT (full attention)	–	50.2	2.6
VGGT sparse (oracle)	yes	59.1	2.1
VGGT sparse (frozen DINO)	no	38.5	3.4
VGGT sparse (FT DINO)	**no**	**55.1**	**2.0**

within **3** of oracle · **+5** vs dense · self-supervised

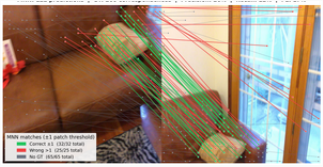
It works because the matches got better

Correspondence F1 (±1)

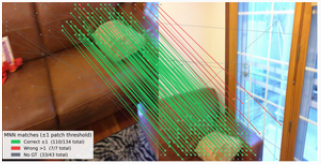
	F1
VGGT sparse (frozen DINO)	32.8
VGGT sparse (FT DINO)	**58.4**

contrastive ≈ **2** ×

frozen DINO



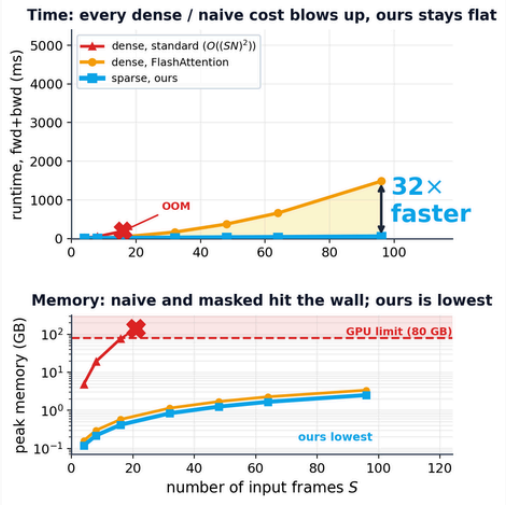
FT DINO



Sparse attention only computes the useful interactions

Correspondances

	Cost	Example
Full attention	$O((NS)^2)$	$(96\times768)^2$ ≈5B
Sparse attention	O(NSK)	96×768×42 ≈3M



4

Conclusions

Motivation > Methodology > Results > Conclusions

Conclusions

1. **Dense attention is largely unnecessary**: Keeping only geometric correspondences preserves, and can even improve, reconstruction quality.

2. **Sparse attention scales:** By exploiting scene sparsity, we reduce quadratic attention to near-linear complexity, achieving 32× faster inference with much lower memory.

3. **Sparse attention is practical:** The model is robust to noisy correspondences, so perfect matching is not required.

Limitations & what is next

Limitations

- ~3 pts short of oracle
- ScanNet indoor

What is next

Limitations & what is next

Limitations

- ~3 pts short of oracle
- ScanNet indoor

What is next

- more data + larger backbones (VGGT-Ω)

Limitations & what is next

Limitations

- ~3 pts short of oracle
- ScanNet indoor

What is next

- more data + larger backbones (VGGT-Ω)

Thank you!

Maria Pilligua · EPFLCVLab