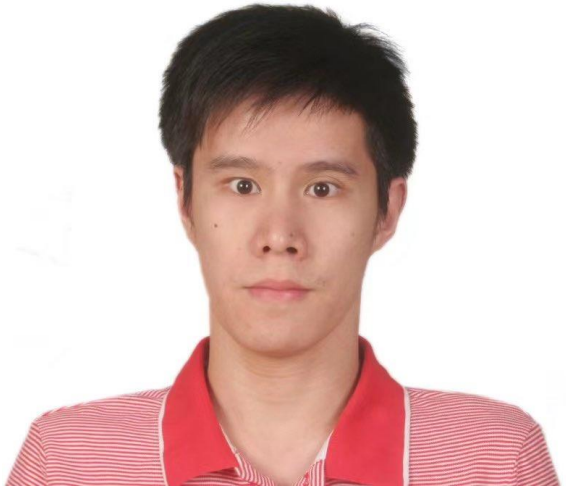
DeDoDe v2: Analyzing and Improving the DeDoDe Keypoint Detector

Johan Edstedt

CVL, Linköping University

Georg Bökman

Chalmers University of Technology



Zhenjun Zhao

Chinese University of Hong Kong

Texas A&M University

Background: 30 Second Intro to DeDoDe v1

- DeDoDe v1 is a keypoint detector
- It's partially trained to detect SIFT detections that survive SfM
  - I.e. 2D detections that are connected to a 3D track
  - This is called the "detection prior"
- It's also trained to maximize self-consistency
  - Done by cross-entropy between single-view logits, and top-k 2-view logits
- It's regularized to spread the detections uniformly over MVS depth

The Good News

v1 ? v2

The Good News

- We made some improvements to DeDoDe:
  - Quantitively:

Mega1500

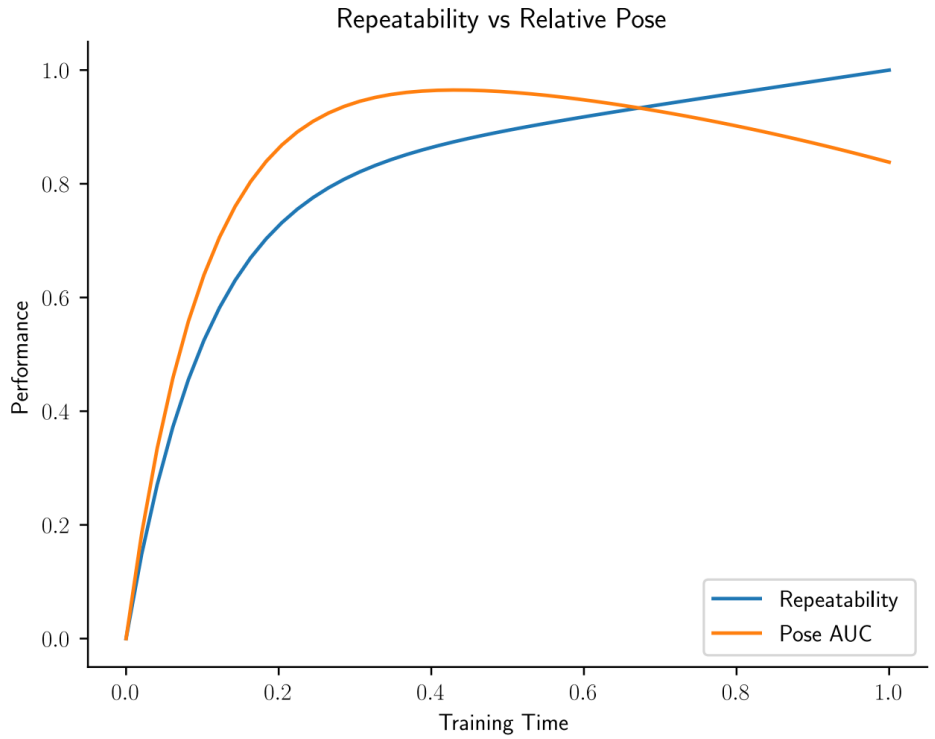
Method ↓ AUC →	@5°	@10°	@20°
SuperPoint [9] CVPRW'18	31.7	46.8	60.1
DISK [30] NeurIps'20	36.7	52.9	65.9
ALIKED [34] TIM'23	41.9	58.4	71.7
SiLK [14] ICCV'23	39.9	55.1	66.9
DeDoDe v1-L — v1-B [12] 3DV'24	49.4	65.5	77.7
DeDoDe **v2**-L — v1-B	**52.5**	**67.4**	**78.7**
DeDoDe C4-L — C4-B [7] CVPR'24	51	67	79
DeDoDe **v2**-L — C4-B	**52.6**	**67.9**	**79.5**
DeDoDe v1-L — v1-G [12] 3DV'24	52.8	69.7	82.0
DeDoDe **v2**-L — v1-G	**54.6**	**70.7**	**82.4**
DeDoDe v1-L — RoMa [13] CVPR'24	55.1	71.6	83.5
DeDoDe **v2**-L — RoMa	**57.6**	**73.3**	**84.4**

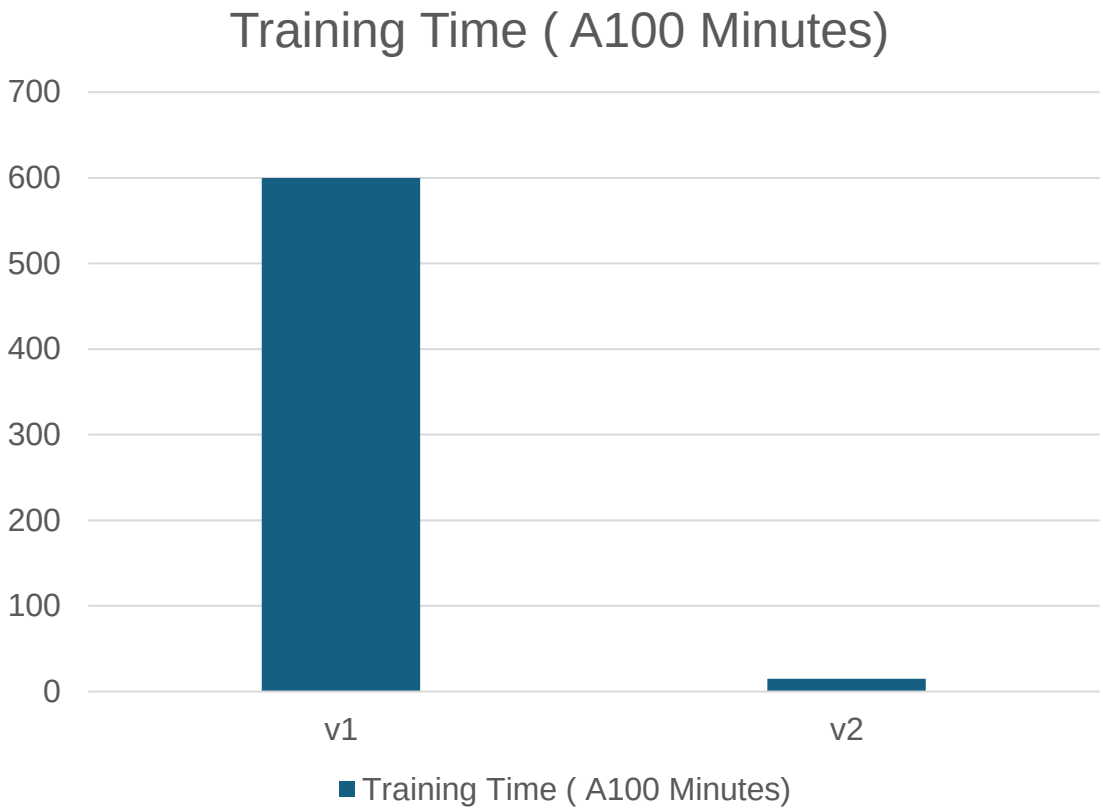
IMC22

Method ↓ mAA →	@10
DISK [30] Neurips'20	64.8
ALIKED [34] IEEE-TIM'23	64.9
SiLK [14] ICCV'23	68.5
DeDoDe v1-L — v1-B [12] 3DV'24	72.9
DeDoDe v2-L — v1-B	**74.7**
DeDoDe v1-L — v1-G [12] 3DV'24	75.8
DeDoDe **v2**-L — v1-G	**78.3**

What's Different?

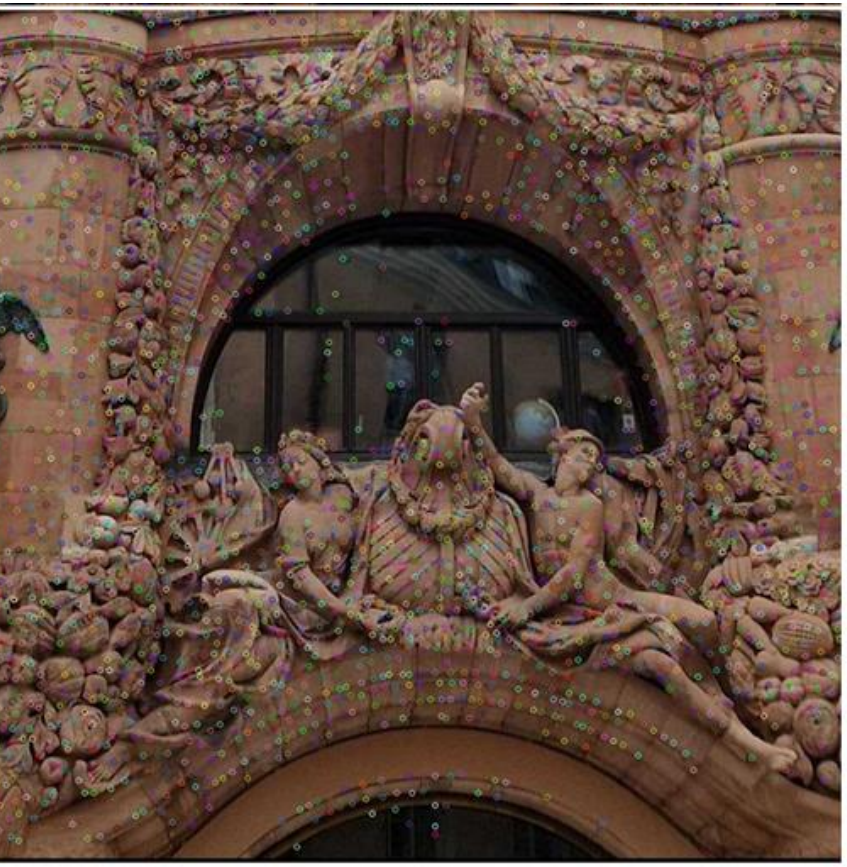
- Significantly shorter training time
  - Prevents overfitting to objective

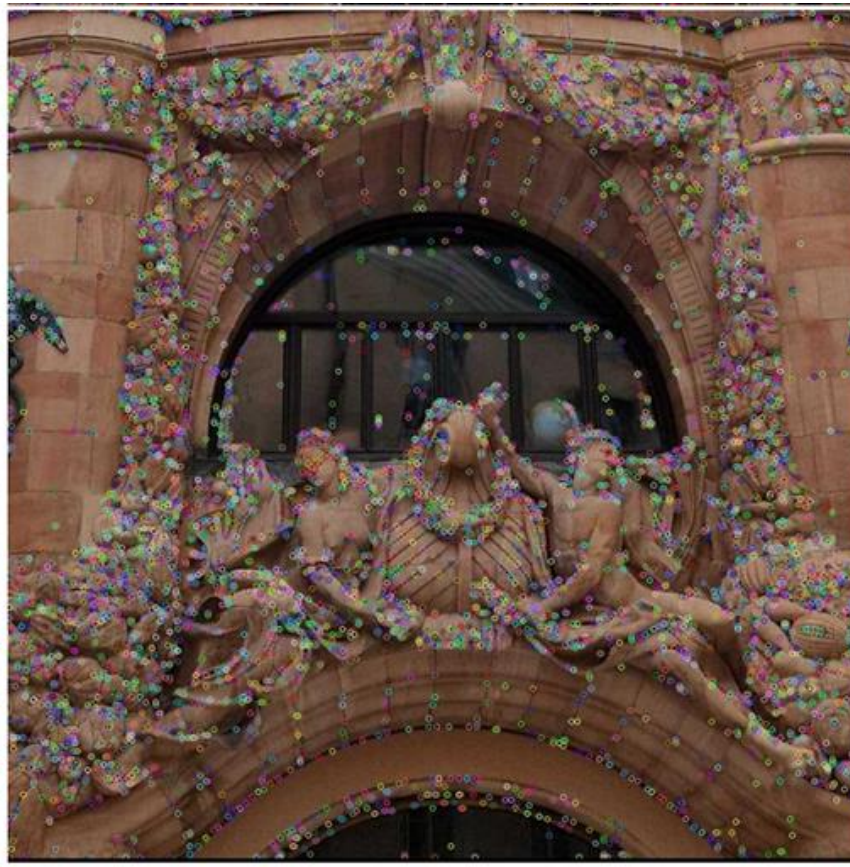




DISK VS DeDoDe v1 VS DeDoDe v2

- DISK not repeatable, but well distributed
- DDD v1 repeatable, not well distributed
- DDD v2 repeatable, ~well distributed





What's Different?

- **Traintime** NMS
  - NMS on targets during training
  - Not during inference (still degrades performance)
  - Similar things of course exist in previous work
    - See e.g. peakiness loss in R2D2

What's Different?

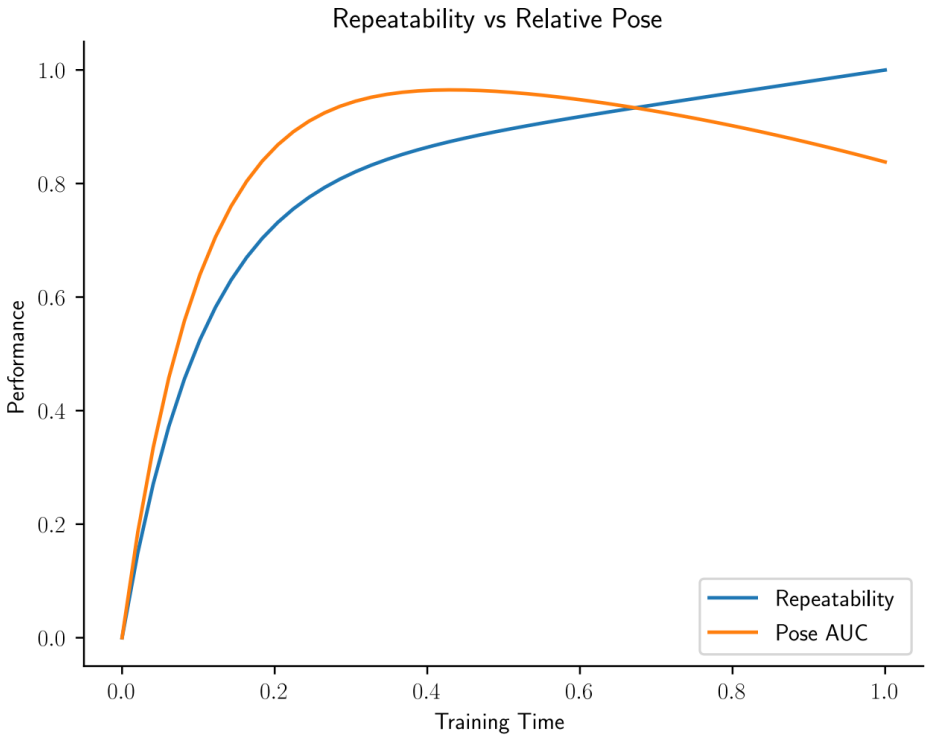
- Minor improvements:
  - Big Rotation Aug
    - Already in Steerers
    - Helps with SO2 invariance
  - Top-k per image instead of per-batch during training
    - Actually this is what is described in v1, but code doesn't do it…
    - Prevents net from ignoring difficult images

What didn’t help?

- A bunch of things we tried with no effect:
  - Post-hoc diversity
    - I.e. trying to modify logits to give better spread
  - Modifying the “detection prior”
    - e.g. the weight (incl. annealing it over training), the standard dev,
  - Modifying the k in top-k
  - Changing the regularizer
    - E.g. to completely uniform, or removing it, or changing the loss weight
  - Learning rate
    - 10^-4 confirmed best learning rate
  - Resolution
    - Investigated lower,higher
    - Also investigated if aspect ratio matters
      - Spoiler: It doesn’t
  - NMS during inference
    - Still doesn’t help us
    - Might not be good in general for “SIFT”y detectors

The Bad News & Some Hope

- Probably repeatability is not such a good metric
  - Needs to be pose, with a good RANSAC
    - Some work in this direction already exists please try it for detectors!



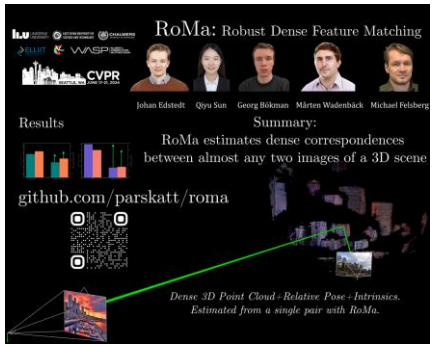
Final Slide

- Weights are in github.com/parskatt/dedode,
- or you can use kornia

```
>>> dedode = DeDoDe.from_pretrained(detector_weights="L-C4-v2", descriptor_weights="B-upright
>>> images = torch.randn(1, 3, 256, 256)
>>> keypoints, scores = dedode.detect(images)
>>> descriptions = dedode.describe(images, keypoints = keypoints)
>>> keypoints, scores, features = dedode(images) # alternatively do both
```

- Also heard it was useful for one of the imc24 teams, happy to hear it ☺
- If you want to get in touch, dm me @ twitter.com/Parskatt
  - Or open an issue and complain about my code, or other stuff I messed up

Final Slide (copy)



- Check out the poster for RoMa!

10:30 - 12:00 **Poster Session 5 & Exhibit Hall** (Arch 4A-E)

25 RoMa: Robust Dense Feature Matching, *Johan Edstedt, Qiyu Sun, Georg Bökman, Mårten Wadenbäck, Michael Felsberg*

FRIDAY, JUNE 21

- And the Oral+Poster for Steerers!

13:00 - 14:30 **Orals 2C: 3D from Multiview and Sensors,** (Summit Flex Hall C)

5 Steerers: A Framework for Rotation Equivariant Keypoint Descriptors, *Georg Bökman, Johan Edstedt, Michael Felsberg, Fredrik Kahl*

WEDNESDAY, JUNE 19

Final Slide (copy 2)

- A keypoint detector detects.

Final Slide (copy 2)

- A keypoint detector detects.
  The points that it detects are called detections.

Final Slide (copy 2)

- A keypoint detector detects.
  The points that it detects are called detections.
- A keypoint descriptor describes.

Final Slide (copy 2)

- A keypoint detector detects.
  The points that it detects are called detections.
- A keypoint descriptor describes.
  The points that it describes are called descriptions.

Final Slide (copy 2)

- A keypoint detector detects.
  The points that it detects are called detections.
- A keypoint descriptor describes.
  The points that it describes are called descriptions.
  - Note that people sometimes misspell these as “descriptors”.

Final Slide (copy 3) (final)

- Will be working on matching again this autumn, looking for group
  - Let me know if you're interested in collaborating