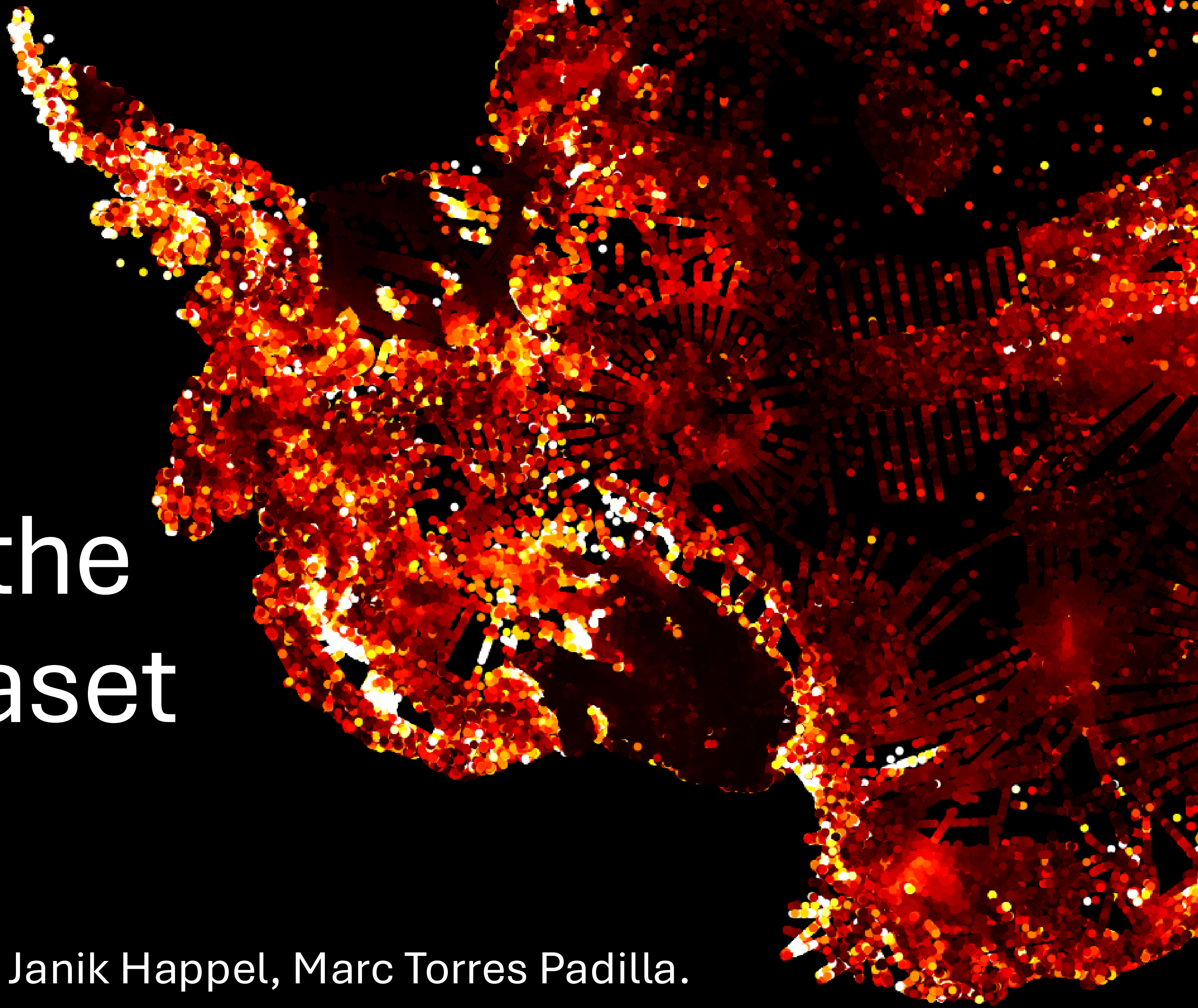


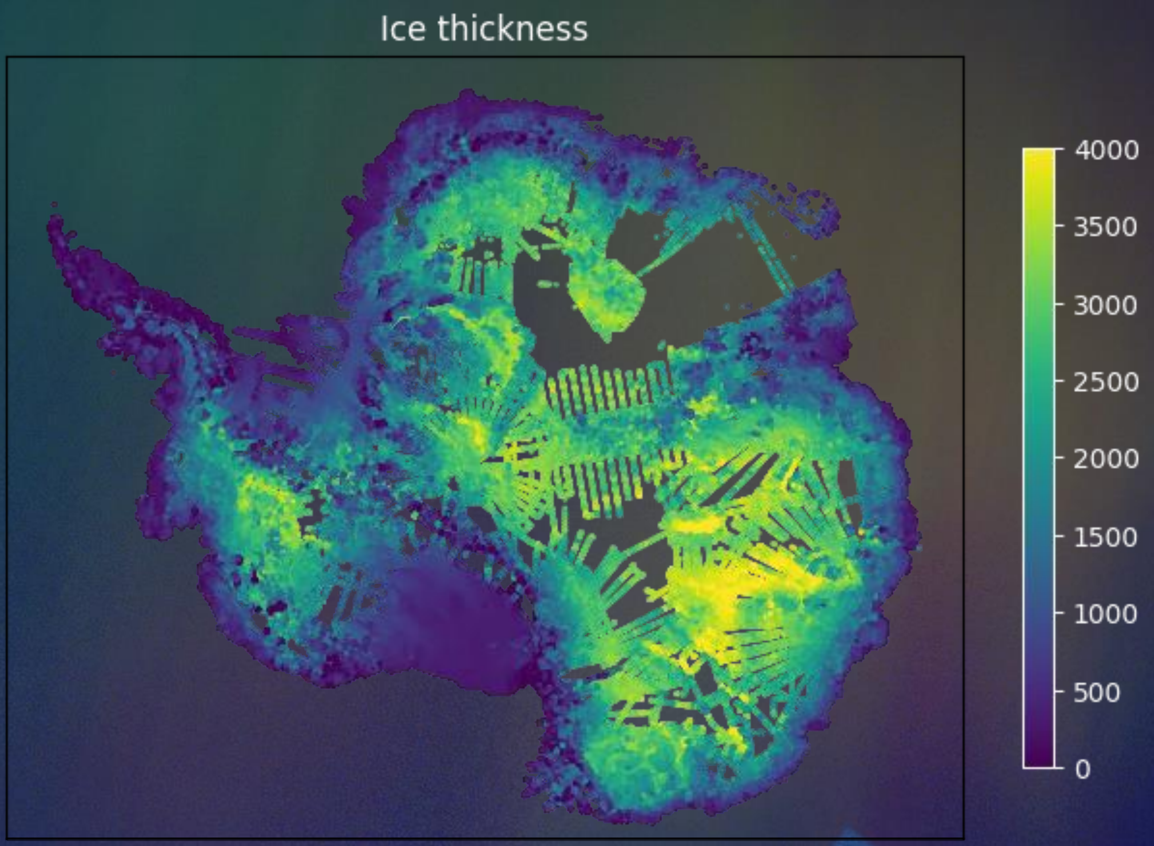
Outlier Detection in the BedMap Dataset

Holger Klevang Christiansen, Janik Happel, Marc Torres Padilla.

University of Copenhagen, Applied Machine Learning 2026

What is BedMap?

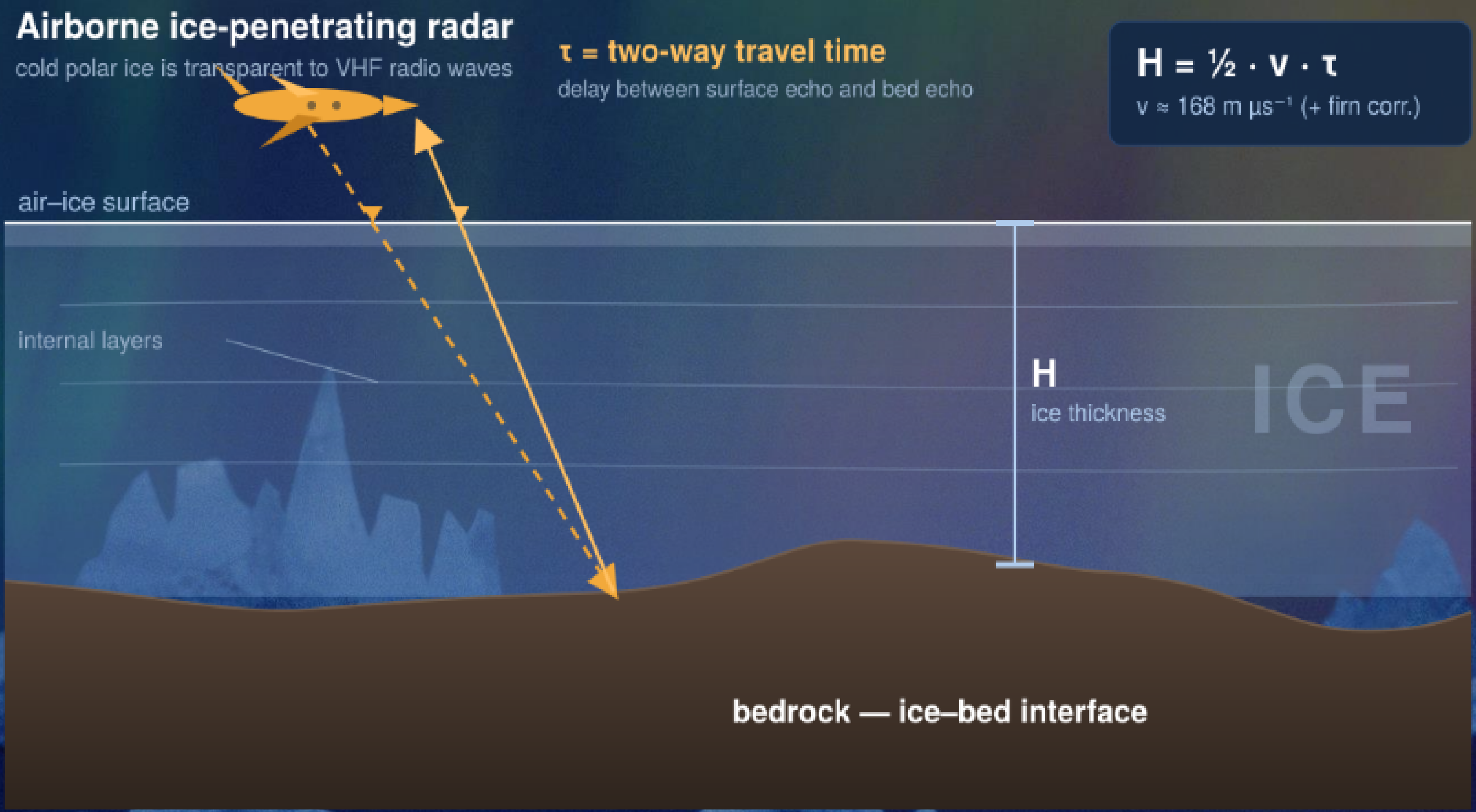
- **Subglacial Topography**— Decades of data (1966-2020)
- **Many variables** — Only 8 are important
- **VHF radio waves** — We measure the echo of the signal



Ice thickness of all the points (in meters)

Foto: © Per-Andre Hoffmann

What is BedMap?



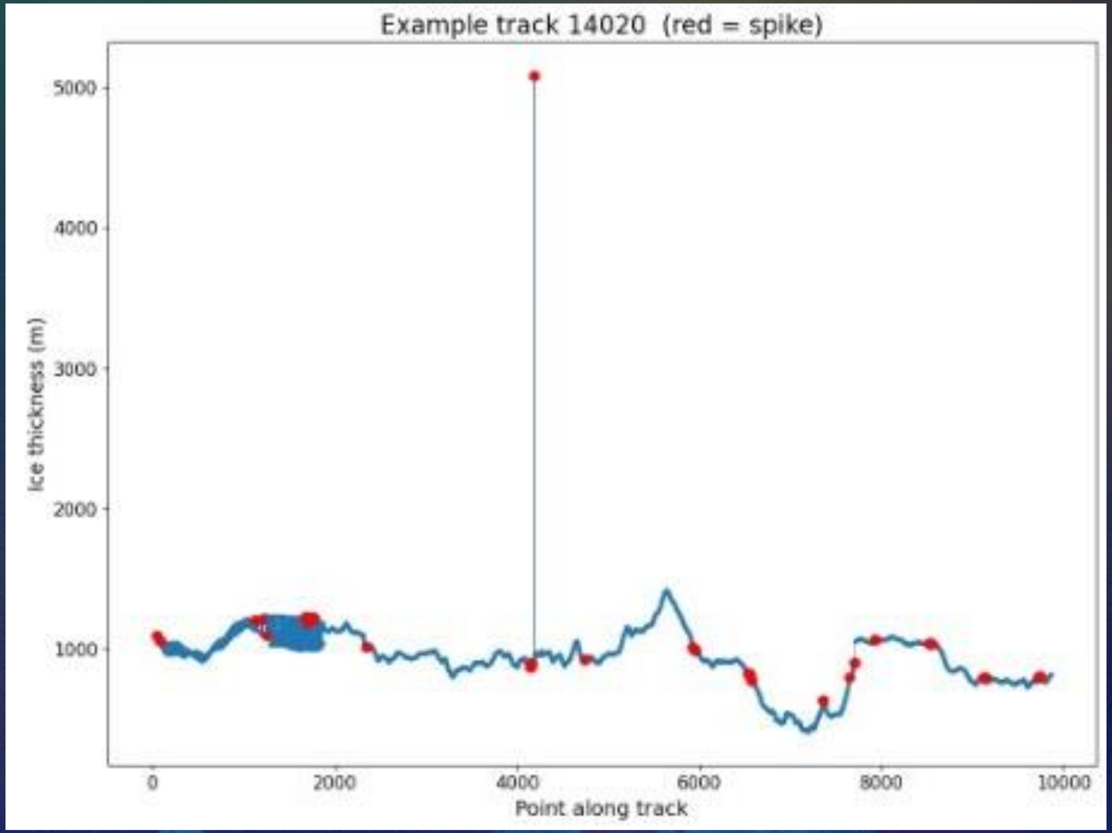
Example of the VHF radio waves method

Foto: © Per-Andre Hoffmann

The problem: uncleaned data

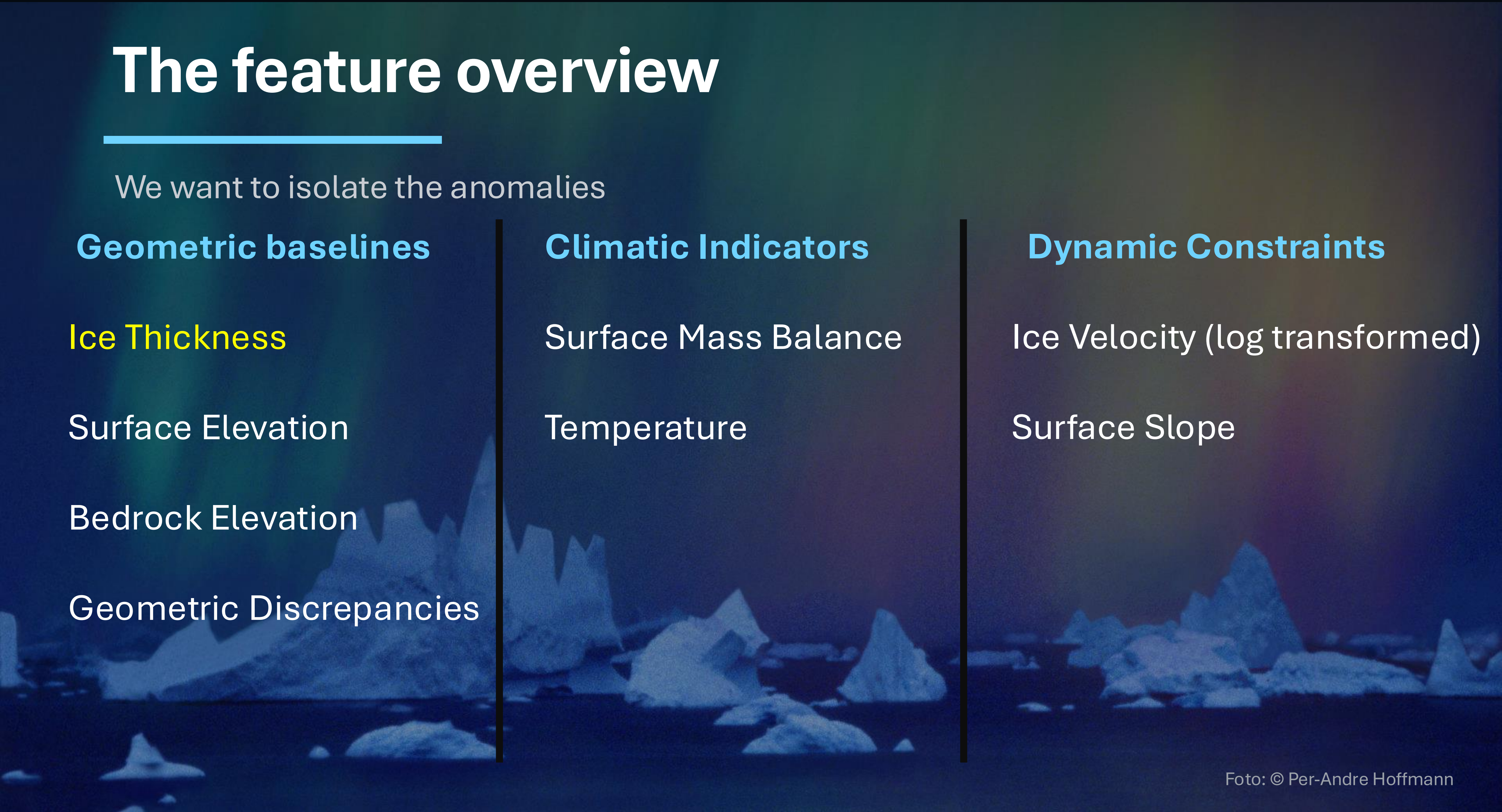
What exactly can be wrong?

- **74.7M points** — In more than 50 years
- **Aleatory noise** — From the hardware and the ice
- **Sudden jumps in thickness** — They will corrupt our modern climate models



Extreme and unphysical jump in a single track

Foto: © Per-Andre Hoffmann



The feature overview

We want to isolate the anomalies

Geometric baselines

- Ice Thickness
- Surface Elevation
- Bedrock Elevation
- Geometric Discrepancies

Climatic Indicators

- Surface Mass Balance
- Temperature

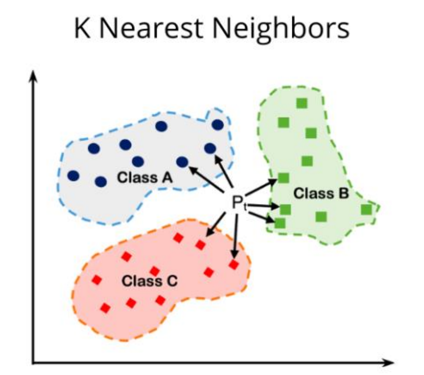
Dynamic Constraints

- Ice Velocity (log transformed)
- Surface Slope

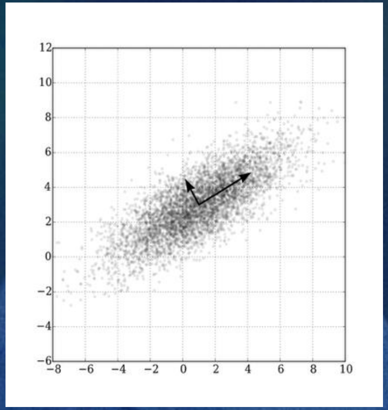
Foto: © Per-Andre Hoffmann

Detection strategy overview

Along-Track-k-Nearest Neighbors

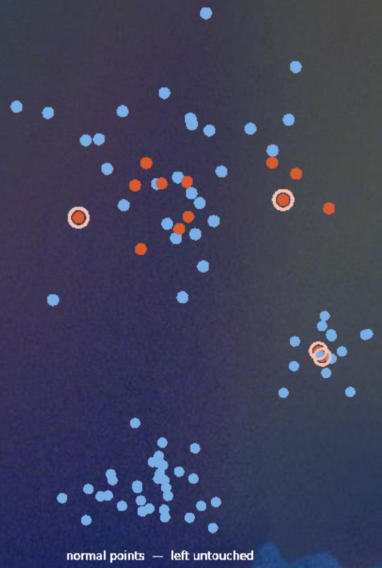


PCA

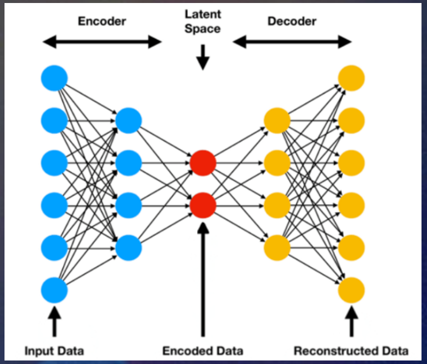


Supervised GNN on Pseudo labels

Latent-space Outlier carving



Graph - Autoencoder

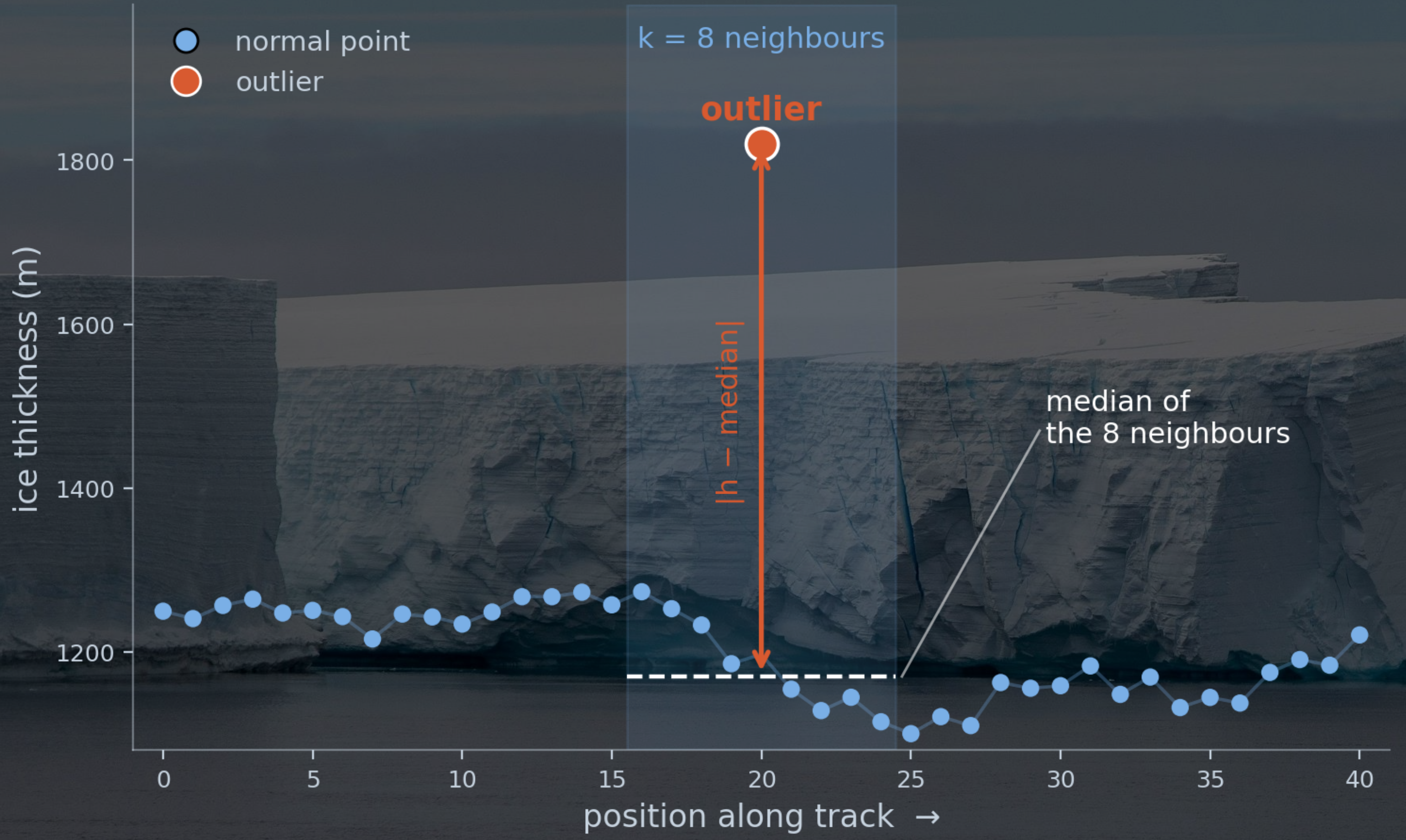


Weighted sum

Foto: © Per-Andre Hoffmann

Along-Track-k-Nearest Neighbours (Spike detection)

Does each point fit its k nearest neighbours along the track?



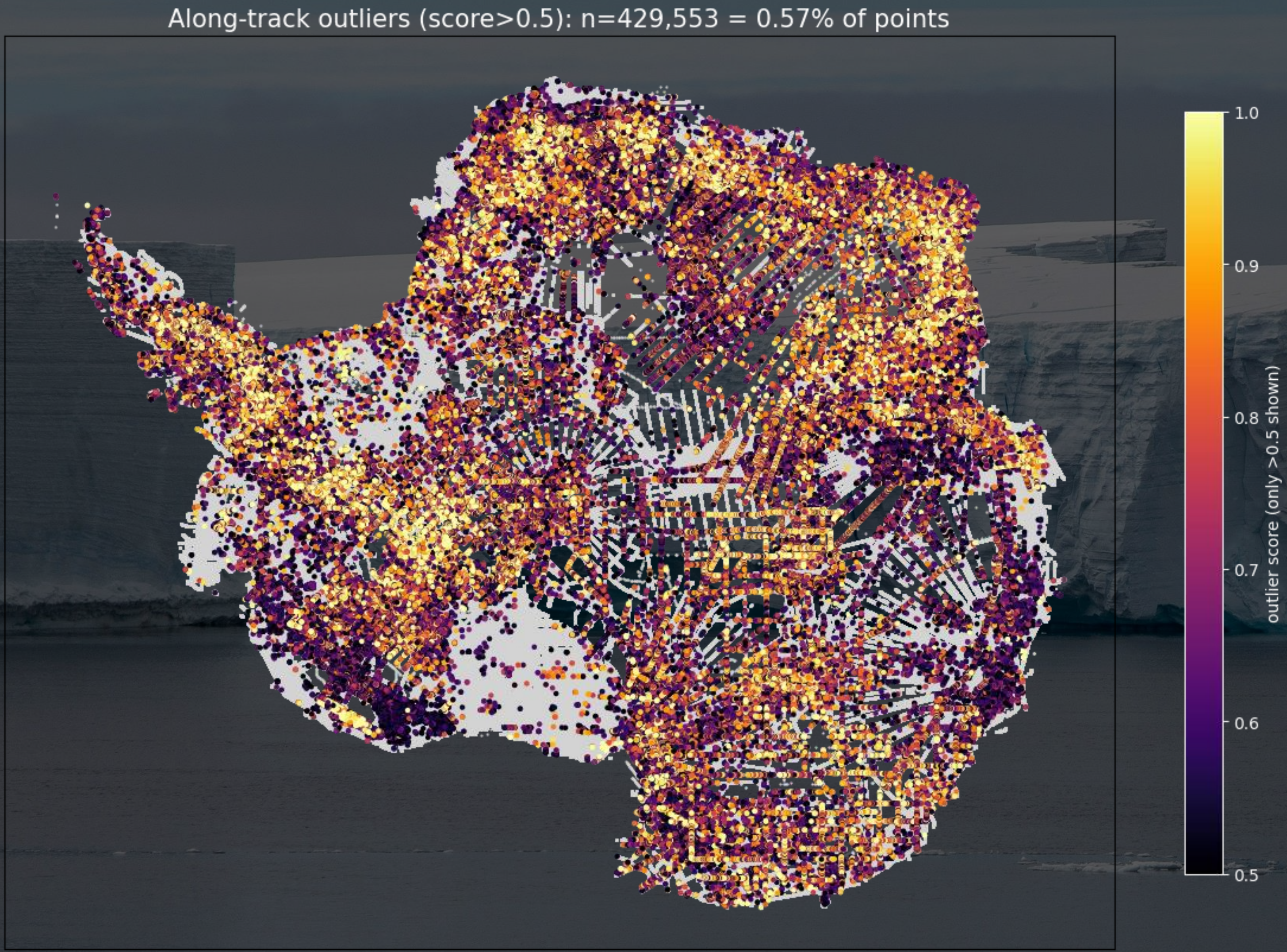
One high score = single spike · many in a row = a bad track sector

1. **Find the 8 nearest neighbours**
   the points just before & after on the track
2. **Reference = their median**
   the mean would let the spike hide itself
3. **Scale by the local IQR**
   adapts to the local terrain roughness
4. **Map through tanh → [0, 1]**
   big deviation → high score → outlier

```
z   =   | h − median | / IQR
score  =  tanh( z )
```

Foto: © ArcticDesire.com Polarreisen, Pexels.com

Along-Track-k-Nearest Neighbours (Spike detection)



Score range	Points	% of all	Median residual (m)
0.00-0.10	69,729,757	93.3 %	0.4
0.10-0.30	4,011,617	5.36 %	14.7
0.30-0.50	544,586	0.729	38.5
0.50-0.70	197,757	0.265	67.0
0.70-0.90	125,464	0.168	108.1
0.90-0.99	67,859	0.091	186.5
0.99-1.00	38,473	0.051	364.0

Foto: © ArcticDesire.com Polarreisen, Pexels.com

Semi-Supervised GNN on Pseudo Labels

- **Overview of the GNN presentation:**

Pseudo-labels

↓

GNN

↓

Results

every point → its 16 nearest neighbours

k-NN-16 map explained in Appendix!

Foto: © Jose Jorquera, AFP/Ritzau Scanpix

1.1 · Candidates for pseudo-labels

Non-ML, pure geometry motivated selection

- **Double hit** — 2 tracks within 25 m, i.e. same spot measured twice (occurs 21.7 million times)
- **Disagree** → outlier; agree → inlier
- **Deciding factor?** :

 - Local support area (2km radius)
 - Depth change between neighbours
 - Slope steepness between neighbours
 - ≥5 tracks, ≥2 surveys, ≥100 points

(short summary version)

A crossover measured twice, independently

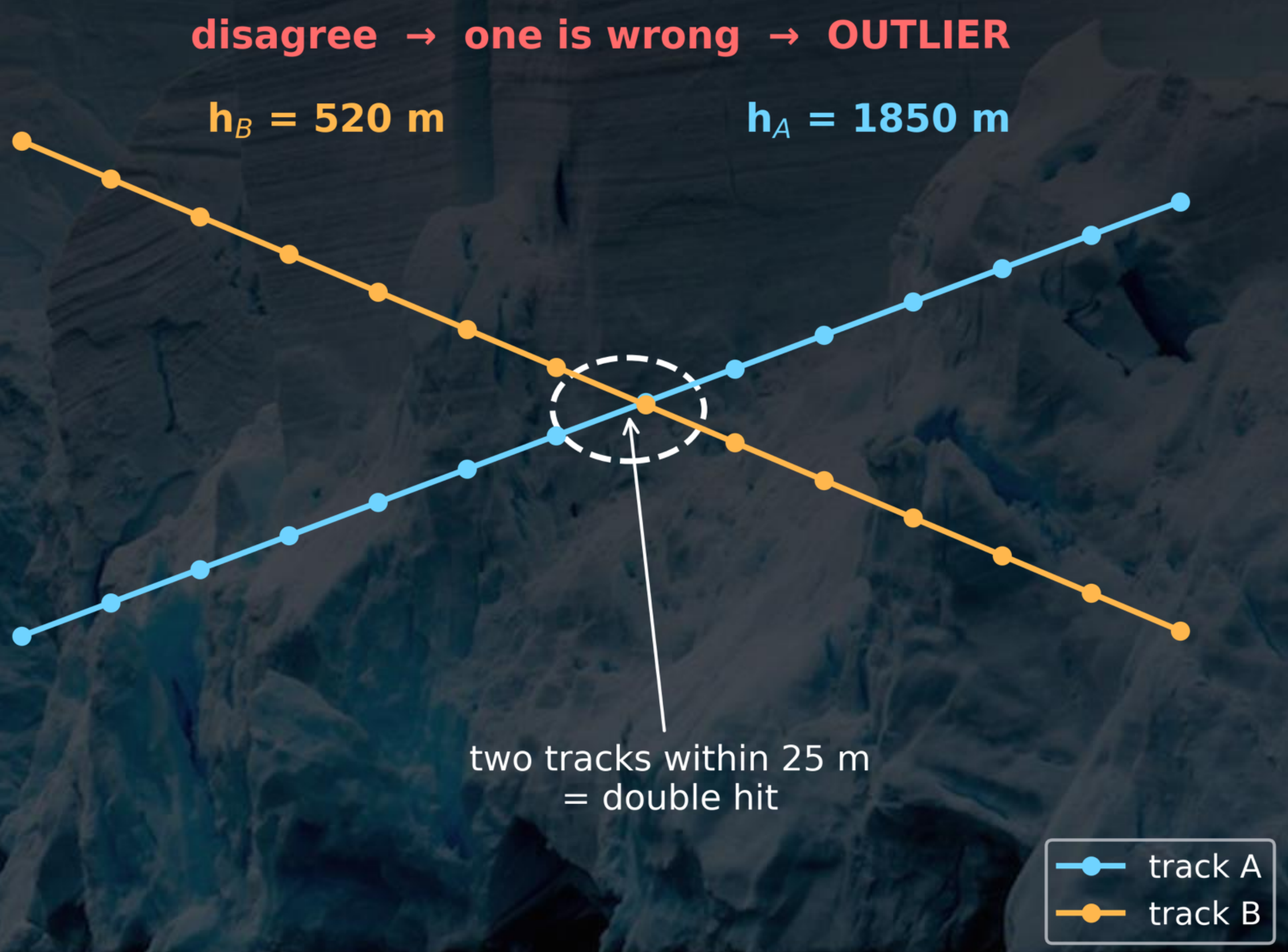


Foto: © Jose Jorquera, AFP/Ritzau Scanpix

1.2 · The resulting pseudo-labels

- **This leaves us with :**

646k unique inliers

39k unique outliers

Ice thickness + outlier seeds

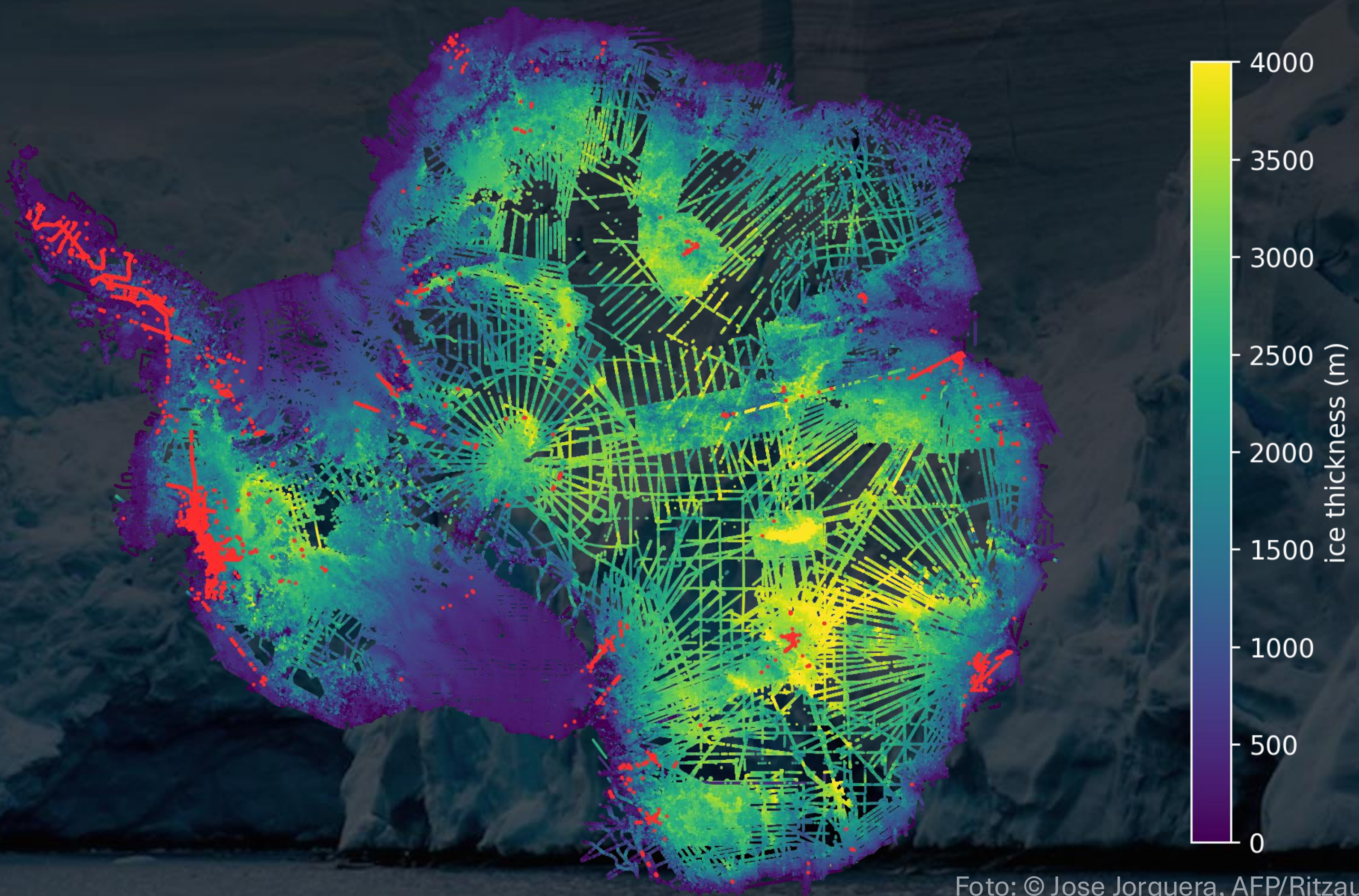


Foto: © Jose Jorquera, AFP/Ritzau Scanpix

3.1· Semi-supervised GNN

- **Input features :**
  Are, what is given in the table, together with the edge table features from last slide

- **Train target :**
  The pseudo truth labels from section 1.

- **Output :**
  Probability of being an outlier given from 0 to 1, where 1 = outlier.

8 raw physics inputs — everything below is a product of these

`ice_thickness · bed_evel · z · s · vx · vy · smb · temp`

11 node features — physics only, each built from the marked inputs

1 `sl_ice = ln(1+ice_thickness)`
2 `bed_evel`
3 `geom_resid = z − bed_evel − ice_thickness`
4 `surf_minus_bed = z − bed_evel`
5 `log_speed = ln(1+√(vx²+vy²))`
6 `s`
7 `z`
8 `smb`
9 `temp`
10 `dir_x = vx / √(vx²+vy²)`
11 `dir_y = vy / √(vx²+vy²)`

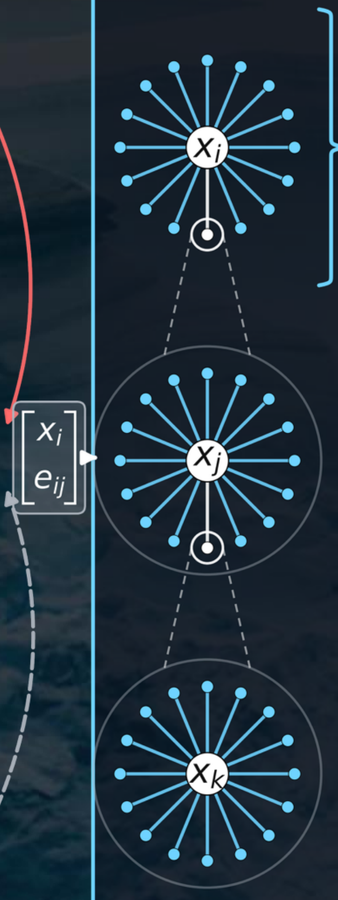
no coords / track / time / BedMachine

Foto: © Jose Jorquera, AFP/Ritzau Scanpix

seed classes

- 🟥 **OUTLIERS**
  label = 1
  38,881
- 🟩 **INLIERS**
  label = 0
  646,674
- ● **unlabeled**
  label = -1
  2,000,000

Message passing



$$m_{ij}(z) = (W_m z) \odot \sigma(W_e e_{ij}) \in \mathbb{R}^{384}$$

$$n_i^{(1)} = \frac{1}{16}\sum_j m_{ij}(x_j) + W_s x_i \in \mathbb{R}^{384}$$

$$n_i^{(2)} = \frac{1}{16}\sum_j m_{ij}(n_j^{(1)}) + W_s n_i^{(1)} \in \mathbb{R}^{384}$$

$$n_i^{(3)} = \frac{1}{16}\sum_j m_{ij}(n_j^{(2)}) + W_s n_i^{(2)} \in \mathbb{R}^{192}$$

$$z_i = [\, n_i^{(1)};\ n_i^{(2)};\ n_i^{(3)} \,] \in \mathbb{R}^{192}$$

MLP head

$\text{logit} = \text{MLP}(z_i)$

$p_{\text{outlier}} = \sigma(\text{logit})$

$\in (0, 1)$

per-point outlier probability

Training loss

weighted cross-entropy + extra unlabeled term:

- 🟥 **outlier · target = 1 → logit ↑**
- 🟩 **inlier · target = 0 → logit ↓**
- ● **unlabeled · target = −1 → logit ↓**

$$\ell^+ = \ln(1 + e^{-\text{logit}}) \qquad \ell^- = \ln(1 + e^{+\text{logit}})$$

$$\mathcal{L} = w_{\text{out}}\langle \ell^+ \rangle_{\text{out}} + w_{\text{in}}\langle \ell^- \rangle_{\text{in}}$$
$$+ w_{\text{unl}}\max(0, \langle \ell^- \rangle_{\text{unl}} - \pi\langle \ell^- \rangle_{\text{out}})$$

$\langle \cdot \rangle$ = *mean over the group's points*

Optuna weights: $w_{\text{out}} = 3.92,\ w_{\text{in}} = 0.996$
$w_{\text{unl}} = 1,\ \pi = 0.0021$

$W:\ W_m,\ W_e,\ W_s$

gradient descent · update W · repeat

Foto: © Jose Jorquera, AFP/Ritzau Scanpix

3.2 · The training

- **2-models are trained:**

To avoid classifying on trained data.

Train/val : 90/10 – outliers

Train/val : 80/20 – Inliers

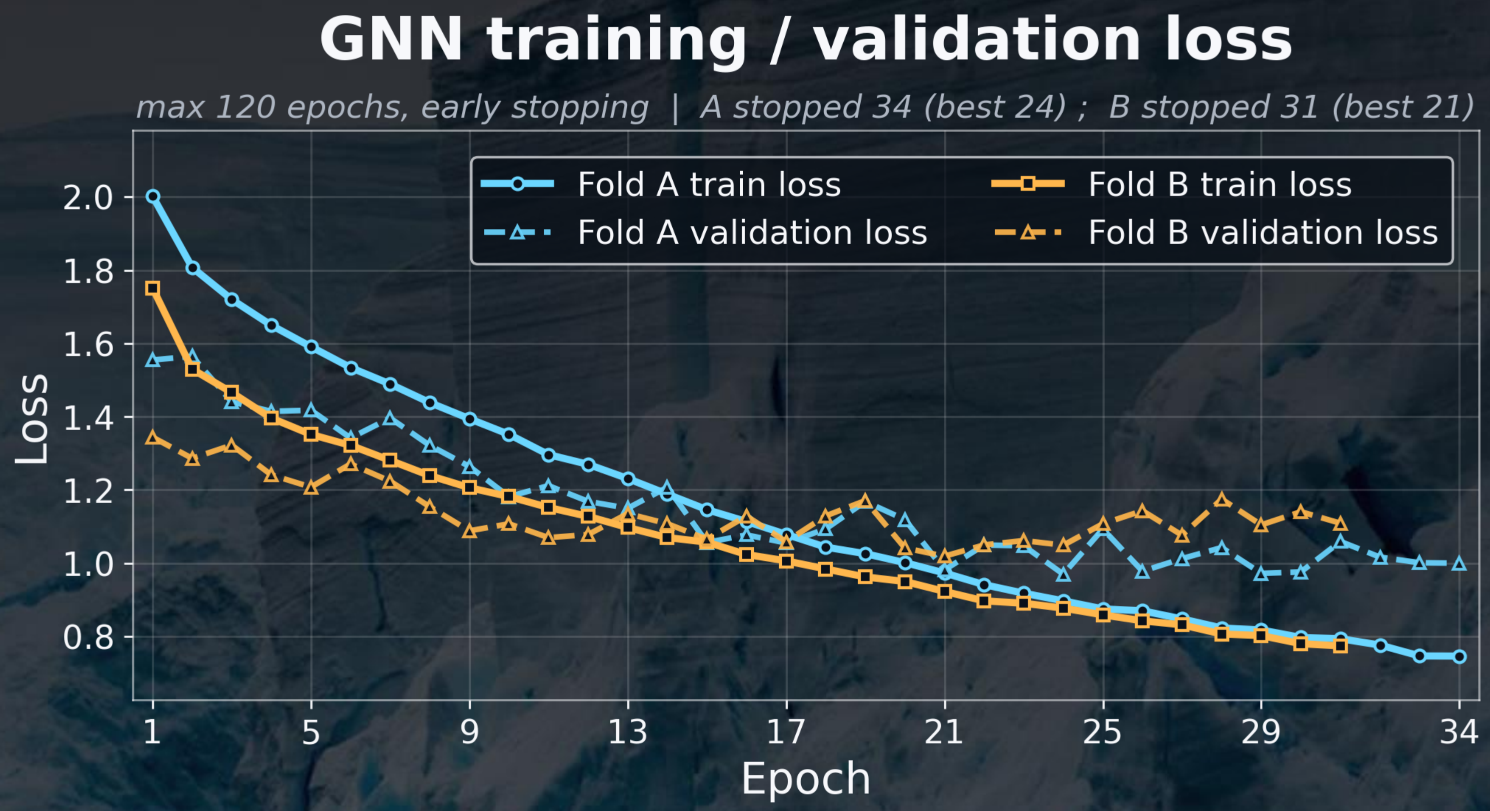
- **Training routine :**

120 epochs with early stopping,

Start lr = 0.00102,

adjusted during run by CosineAnnealingLR,

AdamW optimization for gradient descent



Fold A train nodes:

17,386 outliers, 197,754 inliers

1,000,000 unlabeled

1 hour 34 min, 656k parameters

Fold B train nodes:

17,608 outliers, 319,586 inliers

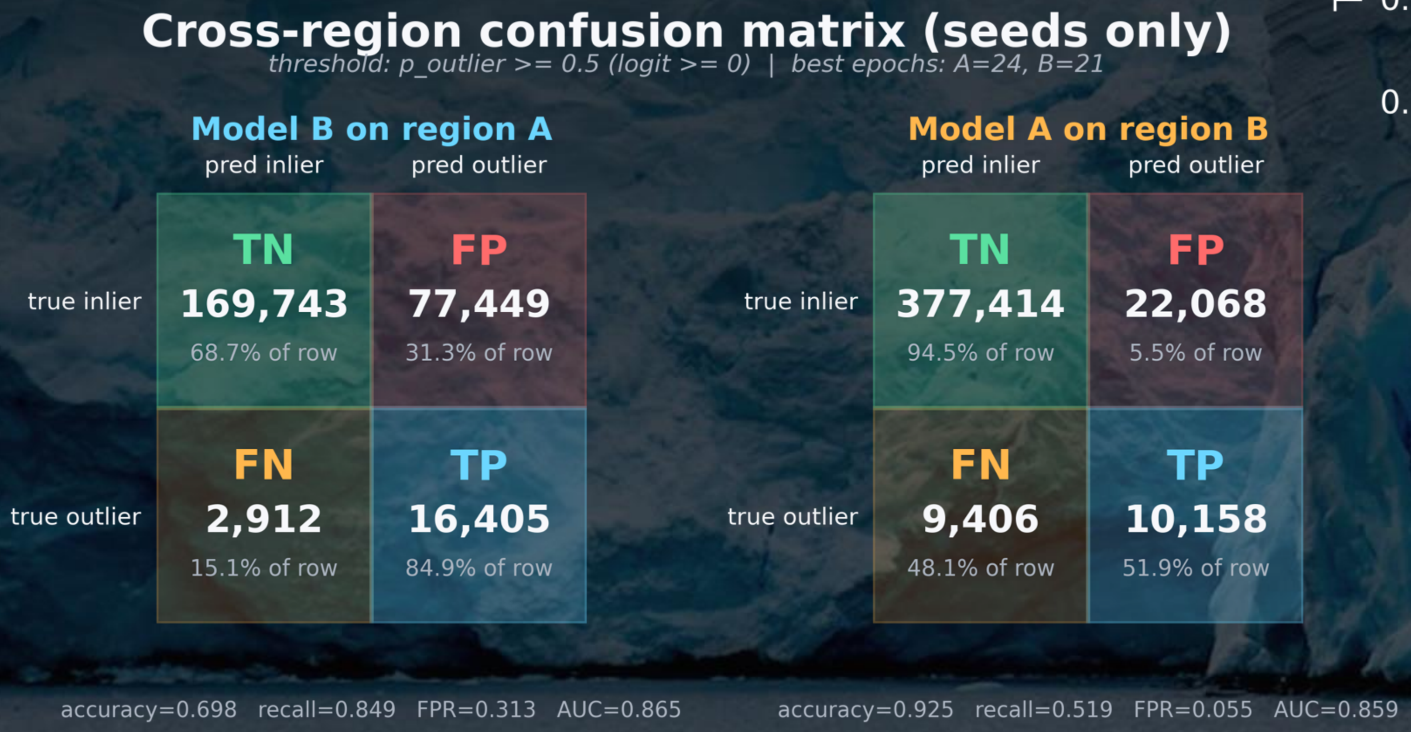
1,000,000 unlabeled

1 hour 54 min, 656k parameters

Foto: © Jose Jorquera, AFP/Ritzau Scanpix

3.3 · GNN Results

- **Cross-region AUC ≈ 0.86** — on the unseen half



Cross-region confusion matrix (seeds only)

threshold: p_outlier >= 0.5 (logit >= 0) | best epochs: A=24, B=21

Model B on region A

	pred inlier	pred outlier
true inlier	TN 169,743 68.7% of row	FP 77,449 31.3% of row
true outlier	FN 2,912 15.1% of row	TP 16,405 84.9% of row

accuracy=0.698 recall=0.849 FPR=0.313 AUC=0.865

Model A on region B

	pred inlier	pred outlier
true inlier	TN 377,414 94.5% of row	FP 22,068 5.5% of row
true outlier	FN 9,406 48.1% of row	TP 10,158 51.9% of row

accuracy=0.925 recall=0.519 FPR=0.055 AUC=0.859

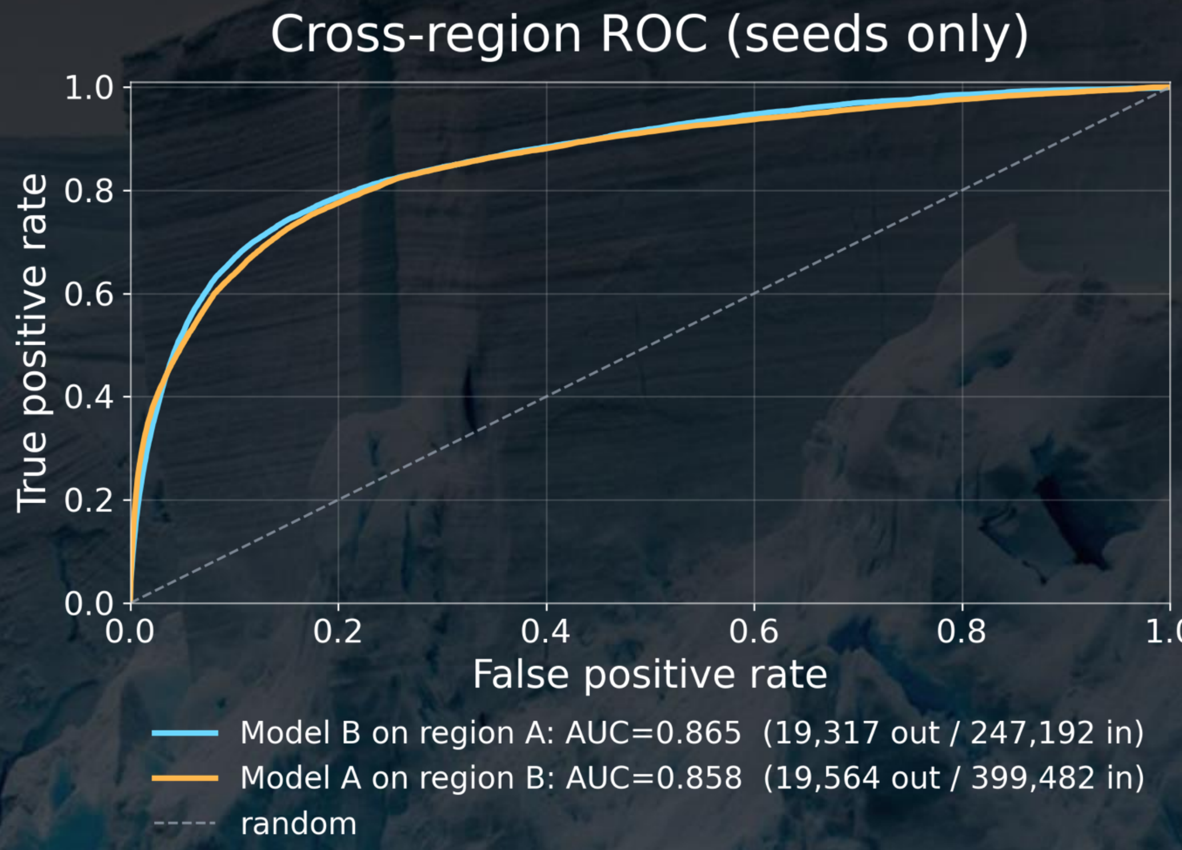


Foto: © Jose Jorquera, AFP/Ritzau Scanpix

3.3 · GNN Results

- **Logit distribution,** is based on the scoring of both models.

38k outliers, 649k inliers.

$logit = 0 \Leftrightarrow p_{out} = 0.5$

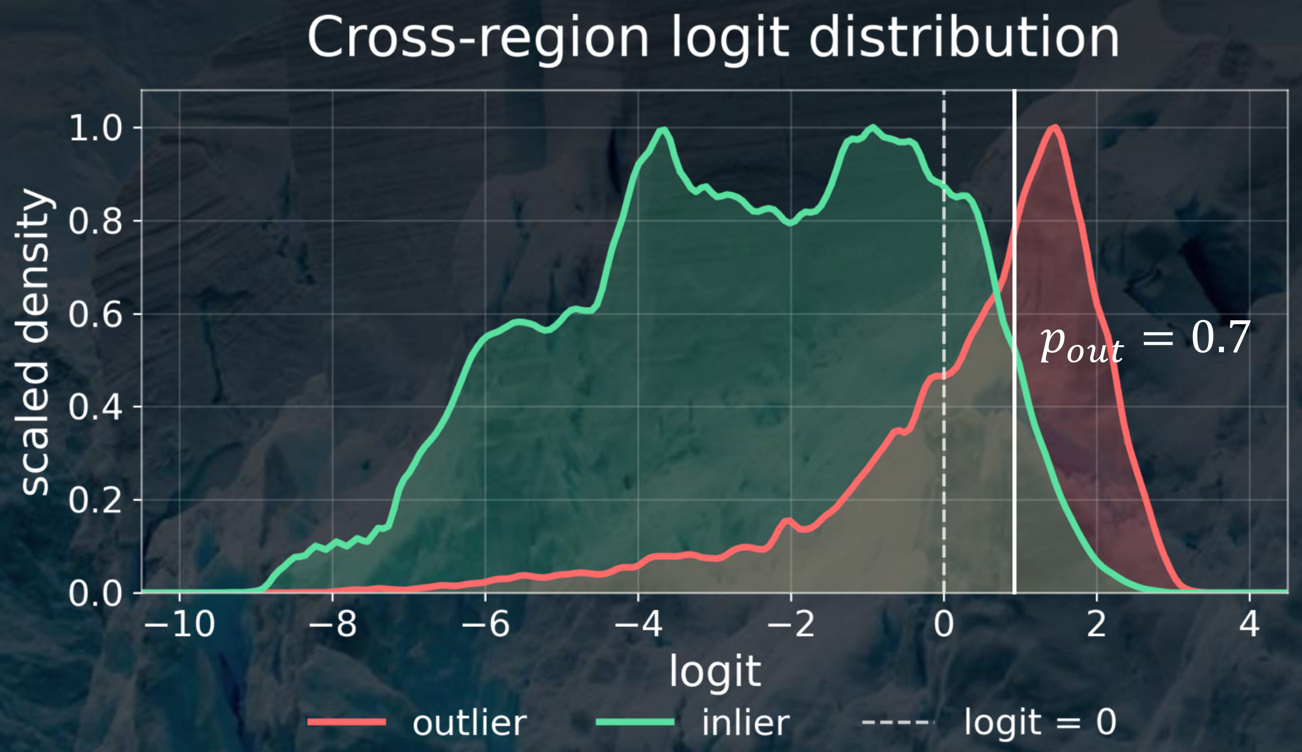


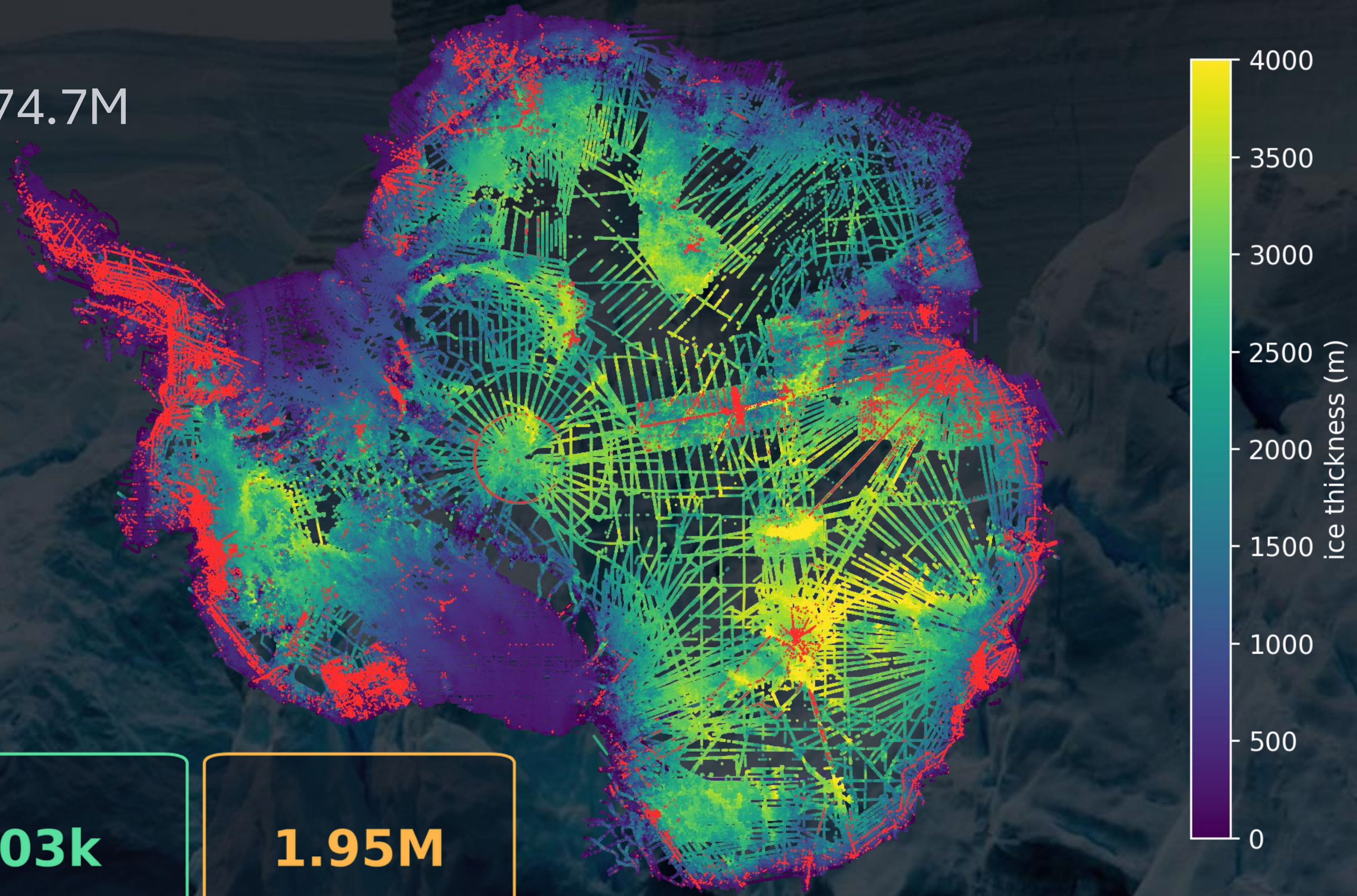
Foto: © Jose Jorquera, AFP/Ritzau Scanpix

3.3 · GNN Results

a point-level outlier score for all 74.7M

- **Outliers** marked by red dots on ice thickness map.
- **Scoring** is done cross fold, between training areas.

Ice thickness + model outliers



AUC ≈ 0.86
cross-region
train one half
predict the other

74.7M
points scored
(cross-fit / out-of-fold)

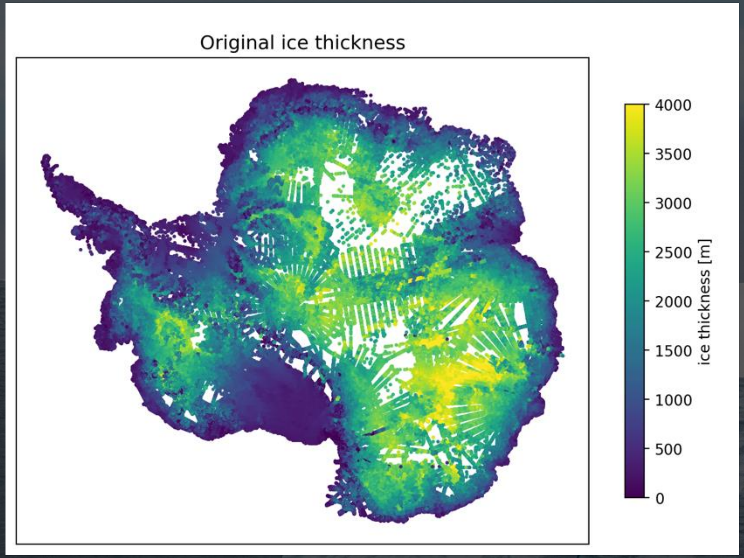
703k
high-confidence
p_outlier > 0.7

1.95M
flagged
p_outlier > 0.5

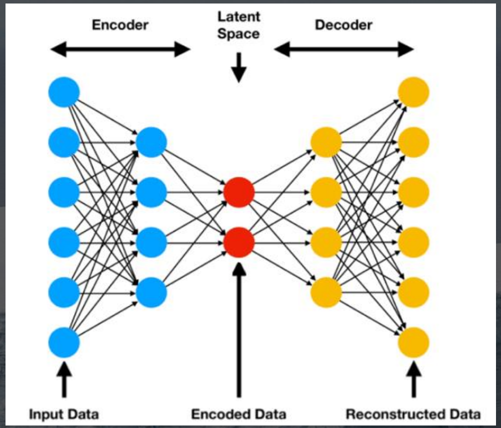
Foto: © Jose Jorquera, AFP/Ritzau Scanpix

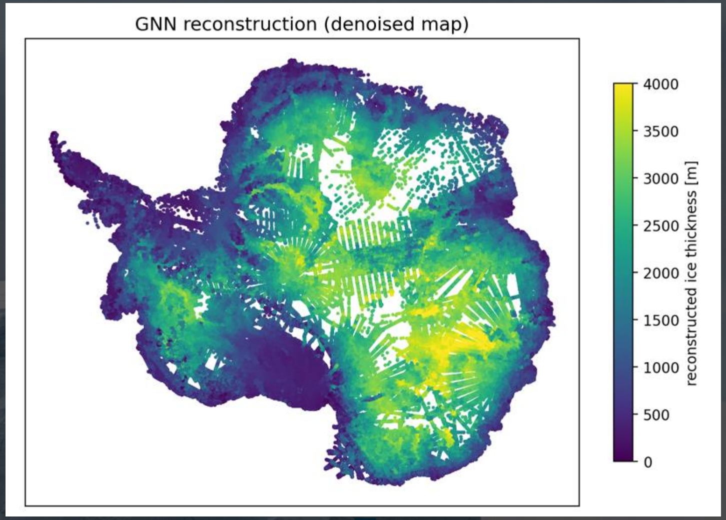
Graph – Autoencoder (AE)

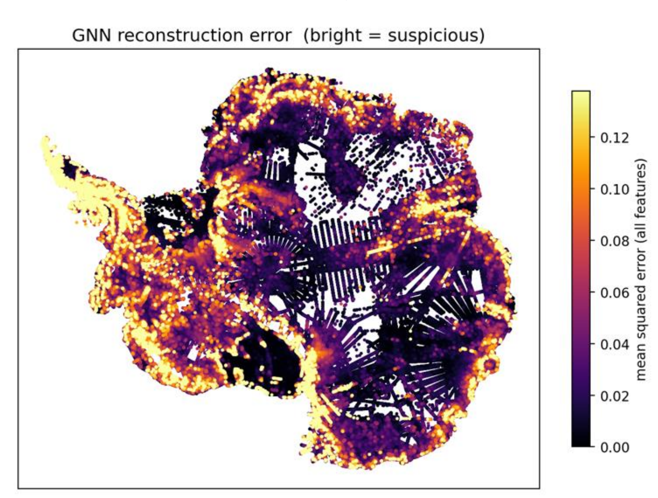
Foto: © ArcticDesire.com Polarreisen, Pexels.com



Autoencoder



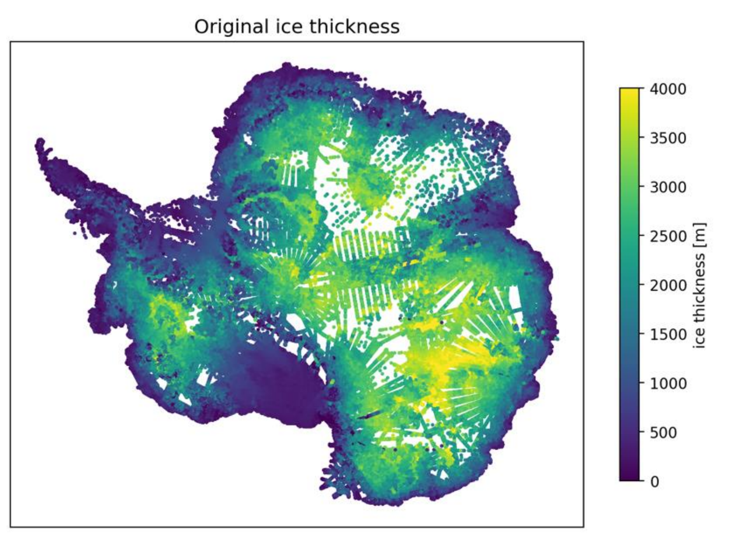




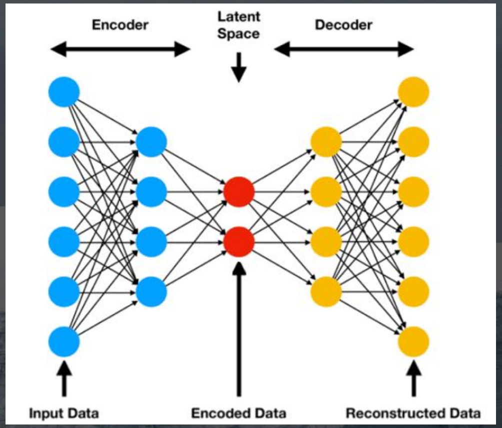
- **Goal:** learn to reconstruct only *normal* ice → outliers can't be reconstructed → **high reconstruction error = outlier**
- **Input features (6):** ice_thickness, z, s_log, smb, temp, v_log
- **Issues of a plain autoencoder:**
    - no **spatial context** — each point is seen alone
    - regresses toward the **mean** → blurry reconstruction
    - **rare-but-real** regions also get high error → false alarms

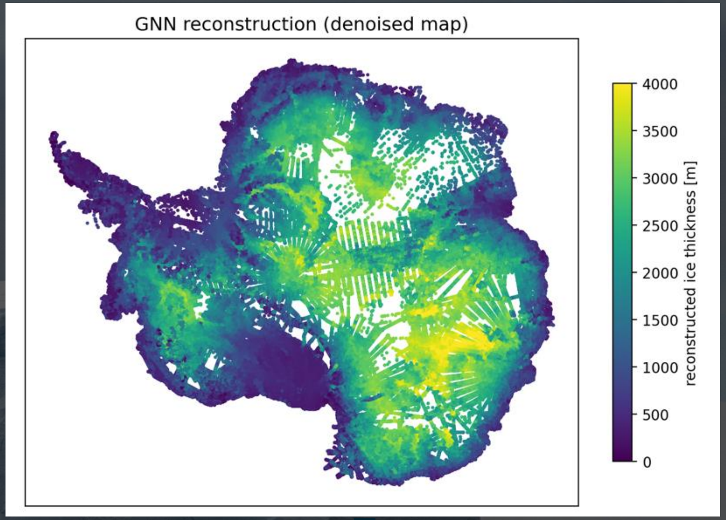
Graph – Autoencoder (AE)

Foto: © ArcticDesire.com Polarreisen, Pexels.com

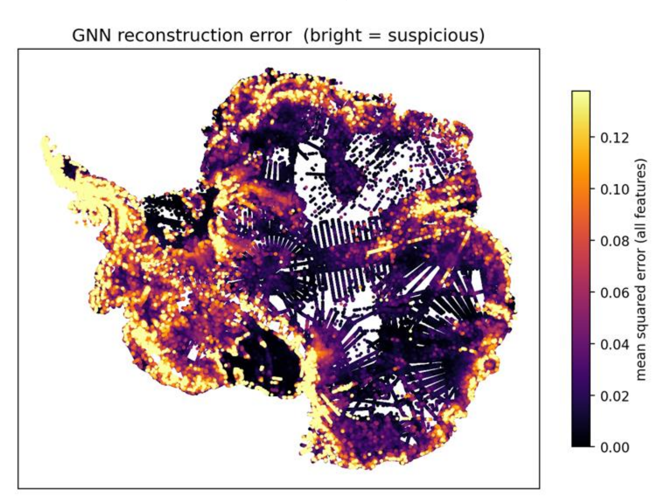

Original ice thickness

Autoencoder




GNN reconstruction (denoised map)

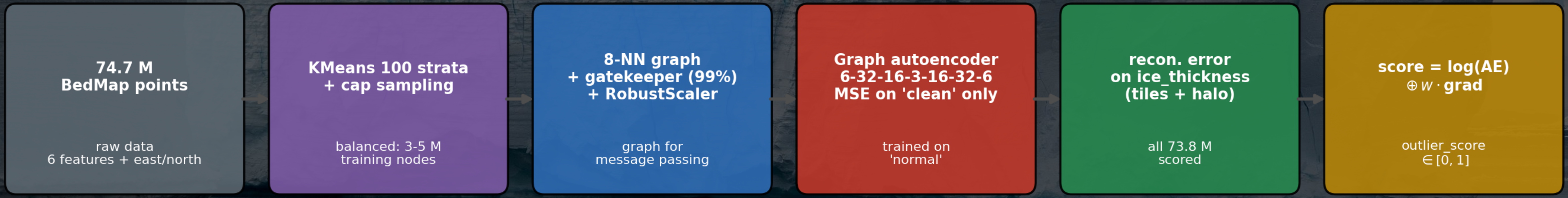
- **Goal:** learn to reconstruct only *normal* ice → outliers can't be reconstructed → **high reconstruction error = outlier**
- **Input features (6):** ice_thickness, z, s_log, smb, temp, v_log
- **Issues of a plain autoencoder:**
  - no **spatial context** — each point is seen alone
  - regresses toward the **mean** → blurry reconstruction
  - **rare-but-real** regions also get high error → false alarms


GNN reconstruction error (bright = suspicious)

Graph – Autoencoder (AE)

End-to-end: from raw point to outlier score

autoencoder output: outlier_score $\in [0, 1]$ *per point*



74.7 M BedMap points
raw data
6 features + east/north

KMeans 100 strata + cap sampling
balanced: 3-5 M training nodes

8-NN graph + gatekeeper (99%) + RobustScaler
graph for message passing

Graph autoencoder 6-32-16-3-16-32-6 MSE on 'clean' only
trained on 'normal'

recon. error on ice_thickness (tiles + halo)
all 73.8 M scored

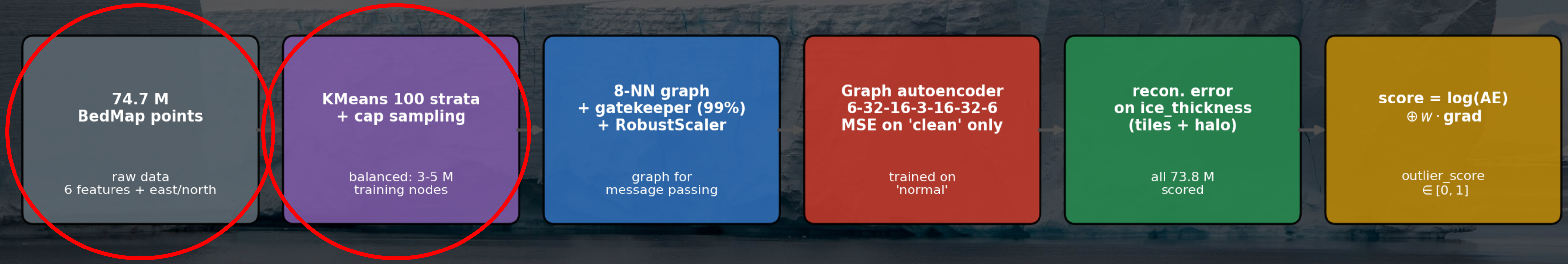
score = log(AE) $\oplus\, w \cdot$ grad
outlier_score $\in [0, 1]$

<- the Graph autoencoder sits here ->

Foto: © ArcticDesire.com Polarreisen, Pexels.com

Graph – Autoencoder (AE)

End-to-end: from raw point to outlier score



autoencoder output: outlier_score $\in [0, 1]$ *per point*

74.7 M BedMap points
raw data
6 features + east/north

KMeans 100 strata + cap sampling
balanced: 3-5 M
training nodes

8-NN graph + gatekeeper (99%) + RobustScaler
graph for
message passing

Graph autoencoder 6-32-16-3-16-32-6 MSE on 'clean' only
trained on
'normal'

recon. error on ice_thickness (tiles + halo)
all 73.8 M
scored

score = log(AE) $\oplus w \cdot$ **grad**
outlier_score
$\in [0, 1]$

<- the Graph autoencoder sits here ->

Foto: © ArcticDesire.com Polarreisen, Pexels.com

Goal: subsample which represents the interer feature space

74.7 M
BedMap points

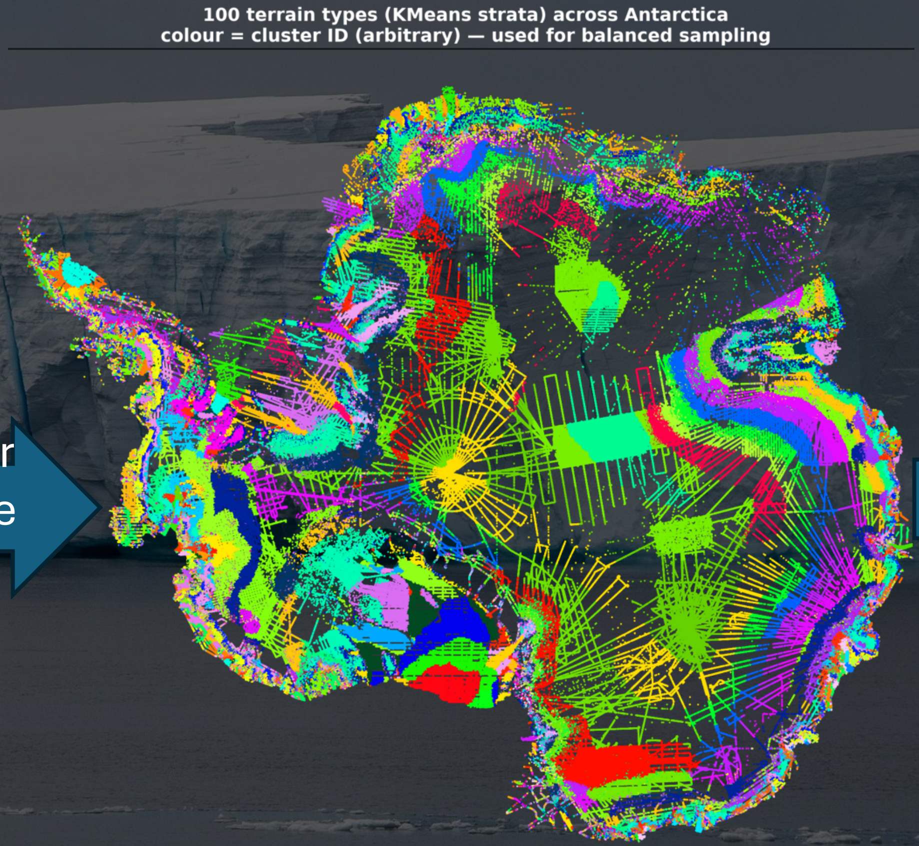
raw data
6 features + east/north

KMeans 100 strata
+ cap sampling

balanced: 3-5 M
training nodes

Physical properties	**Unit**
ice_thickness (surface → bedrock)	m
z (**surface elevation**)	m
s_log (**surface slope**)	
smb (**surface mass balance** (net accumulation))	kg $\mathrm{m}^{-2}\mathrm{yr}^{-1}$
temp (**temperature** (⚠ in **Kelvin**)	K
ice flow speed (magnitude)	m yr^{-1}

Cluster Feature space



100 terrain types (KMeans strata) across Antarctica
colour = cluster ID (arbitrary) — used for balanced sampling

subsample per cluster

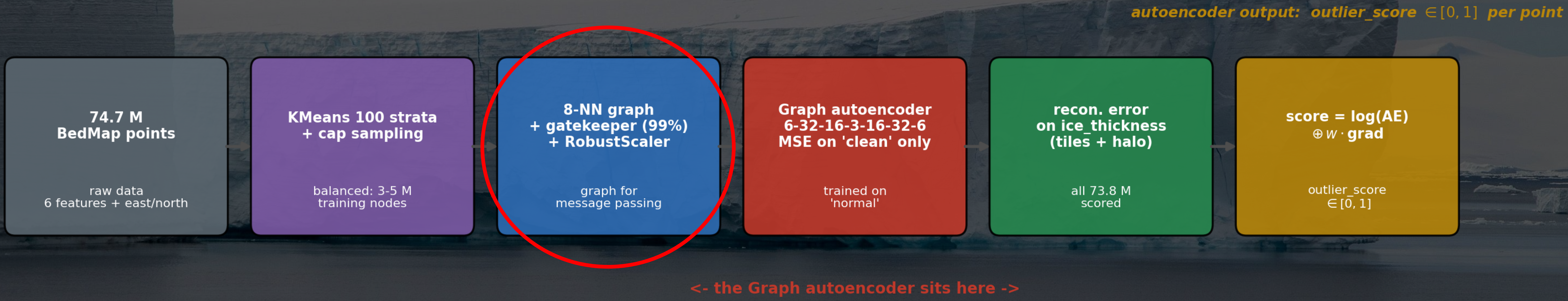
Per Cluster: 50k

100 Clusters * 50k = 5M

Foto: © ArcticDesire.com Polarreisen, Pexels.com

Graph – Autoencoder (AE)

End-to-end: from raw point to outlier score



autoencoder output:* *outlier_score $\in [0, 1]$ ***per point***

74.7 M
BedMap points
raw data
6 features + east/north

KMeans 100 strata
+ cap sampling
balanced: 3-5 M
training nodes

8-NN graph
+ gatekeeper (99%)
+ RobustScaler
graph for
message passing

Graph autoencoder
6-32-16-3-16-32-6
MSE on 'clean' only
trained on
'normal'

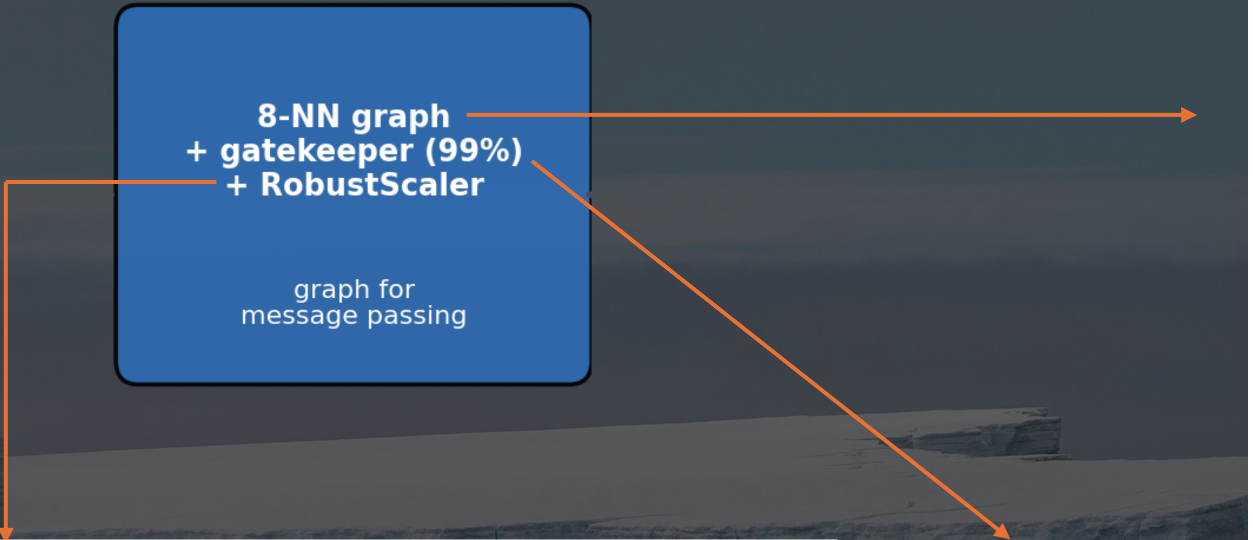
recon. error
on ice_thickness
(tiles + halo)
all 73.8 M
scored

score = log(AE)
$\oplus\, w \cdot$ **grad**
outlier_score
$\in [0, 1]$

<- the Graph autoencoder sits here ->

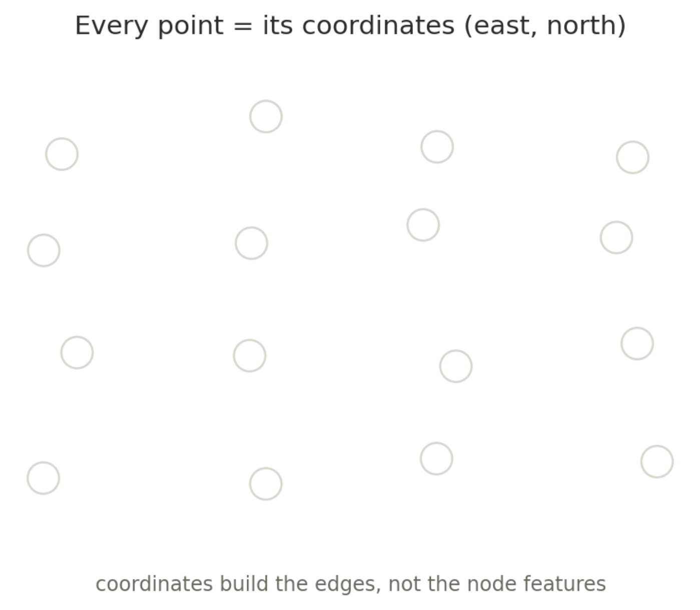
Foto: © ArcticDesire.com Polarreisen, Pexels.com

Foto: © ArcticDesire.com Polarreisen, Pexels.com



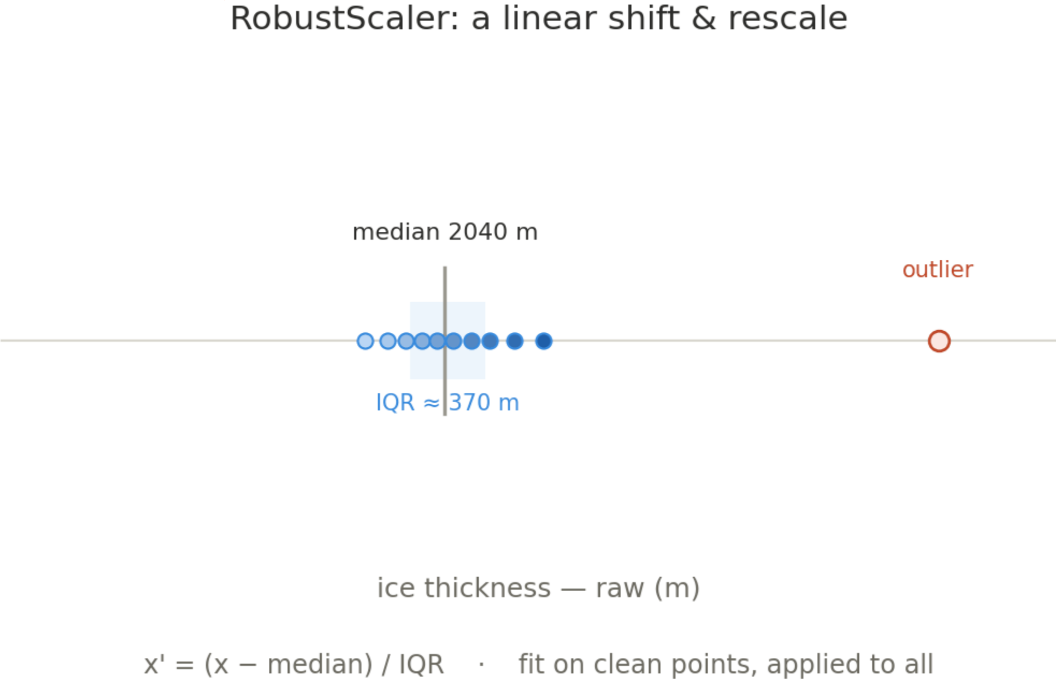
8-NN graph
+ gatekeeper (99%)
+ RobustScaler

graph for
message passing



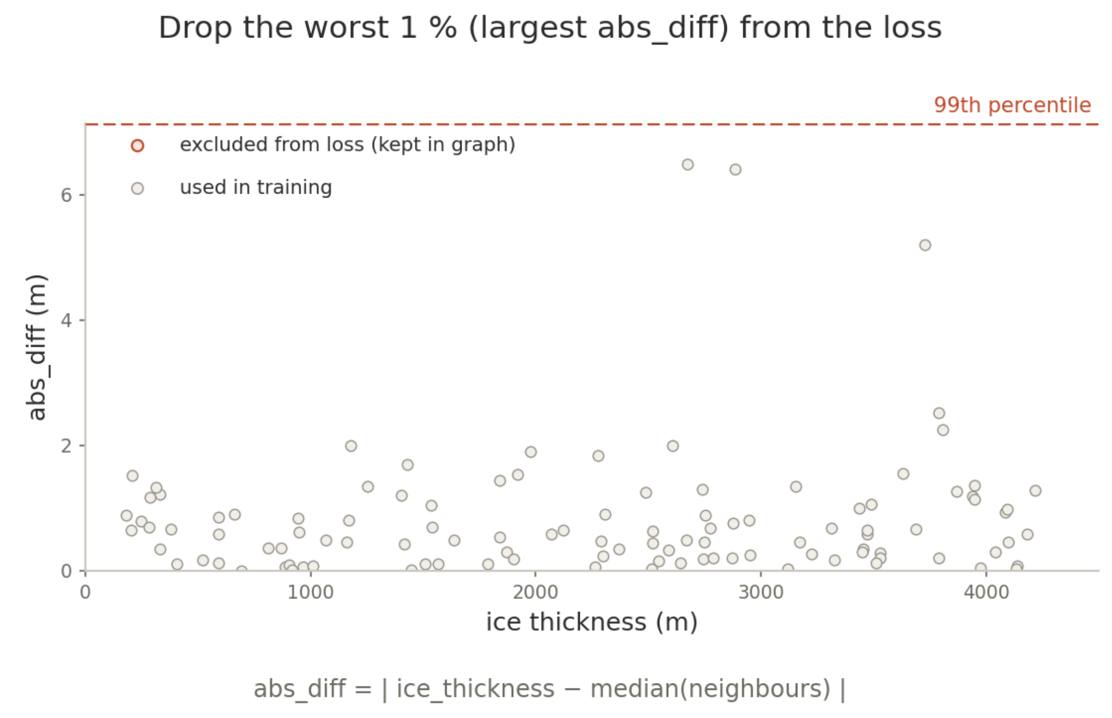
Every point = its coordinates (east, north)

coordinates build the edges, not the node features



RobustScaler: a linear shift & rescale

$x' = (x - \text{median}) / \text{IQR}$ · fit on clean points, applied to all

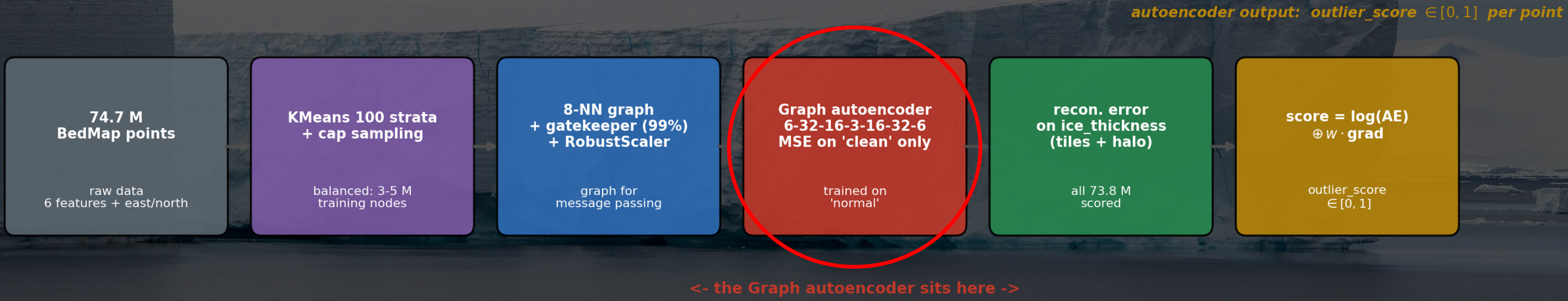


Drop the worst 1 % (largest abs_diff) from the loss

abs_diff = | ice_thickness − median(neighbours) |

Graph – Autoencoder (AE)

End-to-end: from raw point to outlier score



autoencoder output: outlier_score $\in [0, 1]$ ***per point***

74.7 M
BedMap points
raw data
6 features + east/north

KMeans 100 strata
+ cap sampling
balanced: 3-5 M
training nodes

8-NN graph
+ gatekeeper (99%)
+ RobustScaler
graph for
message passing

Graph autoencoder
6-32-16-3-16-32-6
MSE on 'clean' only
trained on
'normal'

recon. error
on ice_thickness
(tiles + halo)
all 73.8 M
scored

score = log(AE)
$\oplus\, w \cdot$ **grad**
outlier_score
$\in [0, 1]$

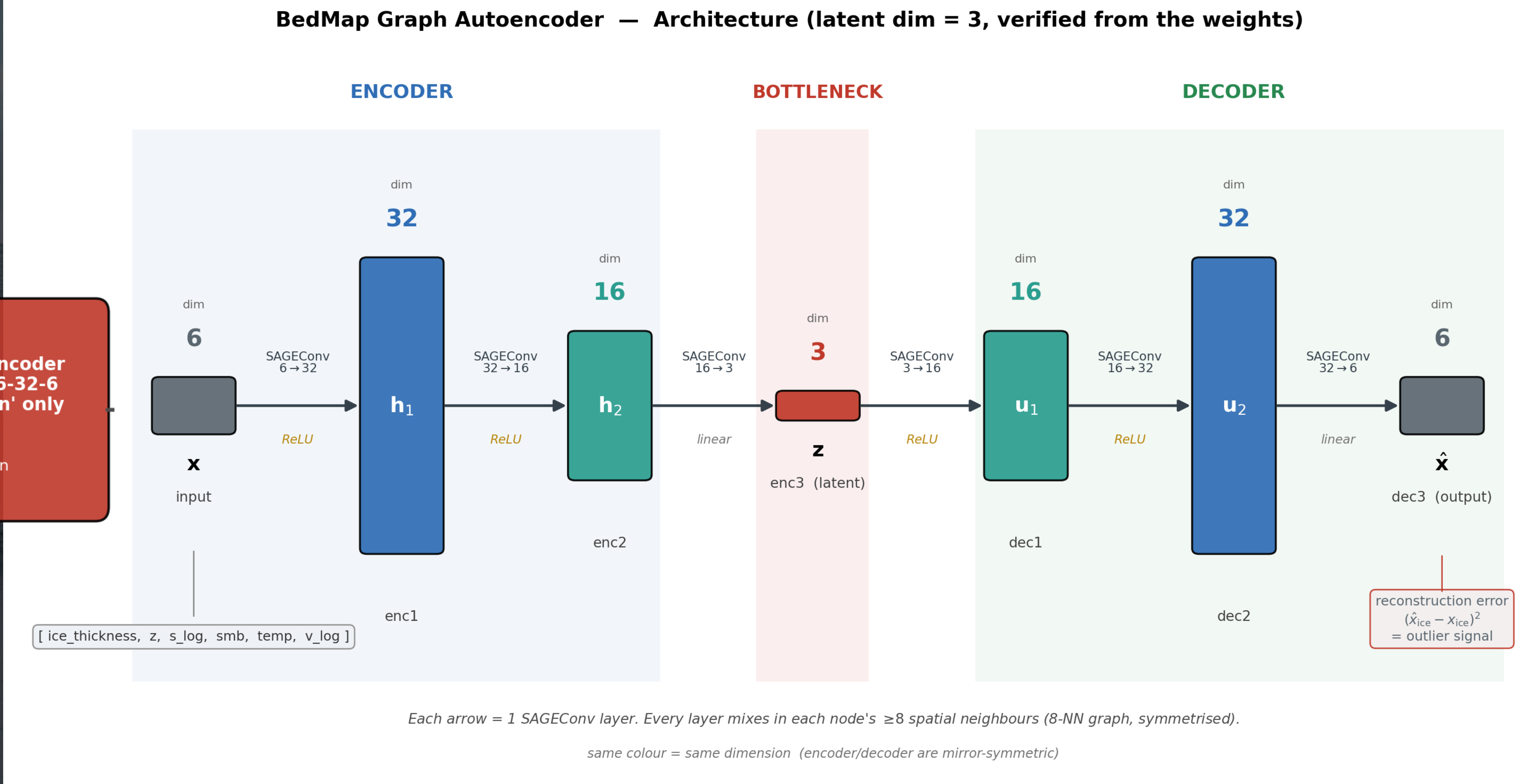
<- the Graph autoencoder sits here ->

Foto: © ArcticDesire.com Polarreisen, Pexels.com

Graph – Autoencoder (AE)

Graph autoencoder
6-32-16-3-16-32-6
MSE on 'clean' only

trained on
'normal'



BedMap Graph Autoencoder — Architecture (latent dim = 3, verified from the weights)

Each arrow = 1 SAGEConv layer. Every layer mixes in each node's ≥8 spatial neighbours (8-NN graph, symmetrised).

same colour = same dimension (encoder/decoder are mirror-symmetric)

Foto: © ArcticDesire.com Polarreisen, Pexels.com

Graph – Autoencoder (AE)

recon. error on ice_thickness (tiles + halo)

all 73.8 M scored

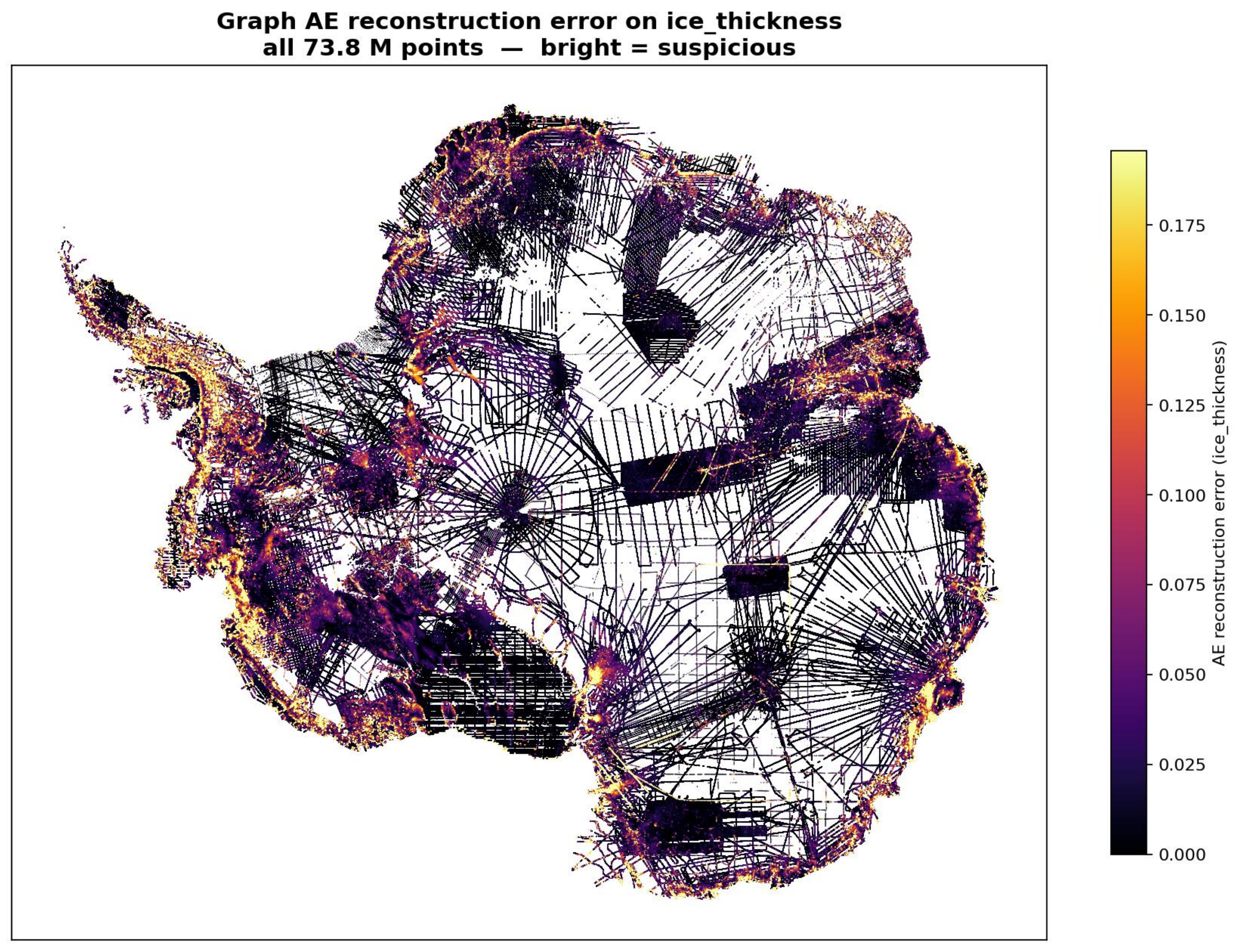
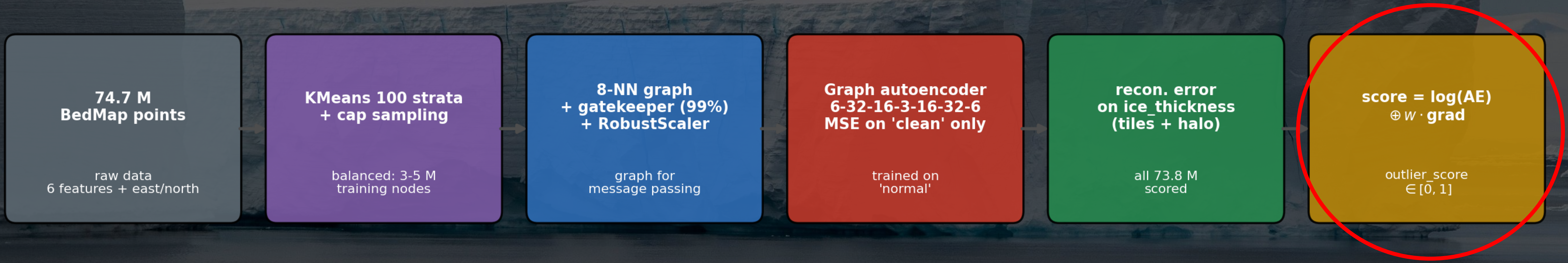


Foto: © ArcticDesire.com Polarreisen, Pexels.com

Graph – Autoencoder (AE)

End-to-end: from raw point to outlier score

autoencoder output: *outlier_score* $\in [0, 1]$ *per point*



74.7 M
BedMap points
raw data
6 features + east/north

KMeans 100 strata
+ cap sampling
balanced: 3-5 M
training nodes

8-NN graph
+ gatekeeper (99%)
+ RobustScaler
graph for
message passing

Graph autoencoder
6-32-16-3-16-32-6
MSE on 'clean' only
trained on
'normal'

recon. error
on ice_thickness
(tiles + halo)
all 73.8 M
scored

score = log(AE)
$\oplus\, w \cdot$ grad
outlier_score
$\in [0, 1]$

<- the Graph autoencoder sits here ->

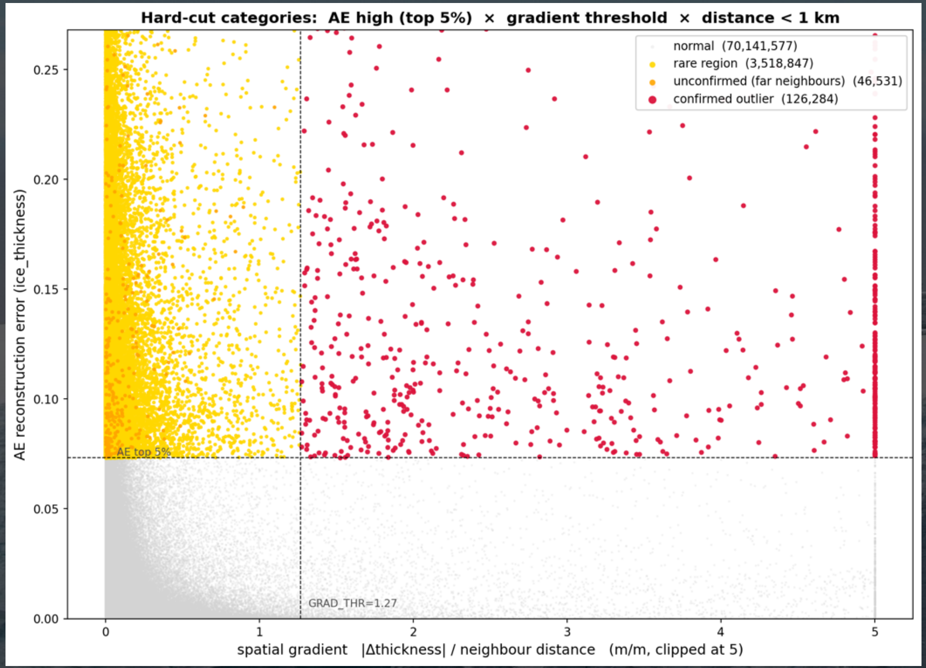
Foto: © ArcticDesire.com Polarreisen, Pexels.com

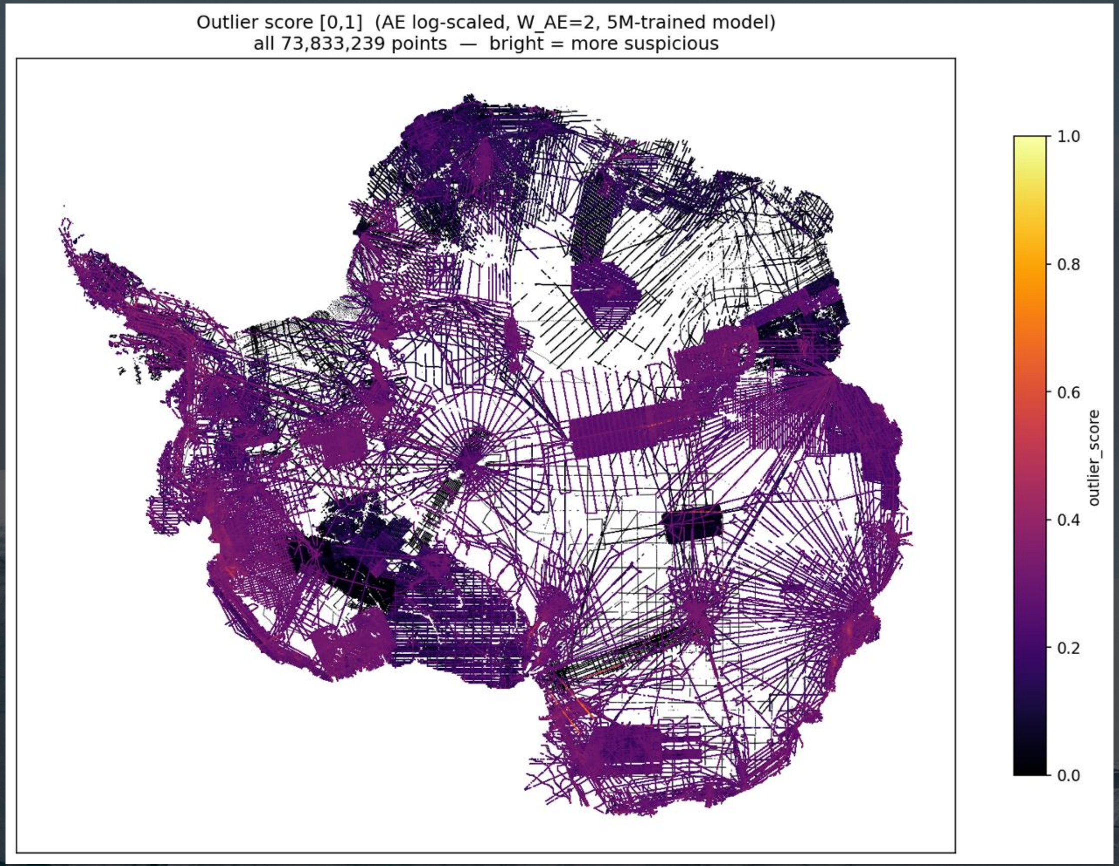
Graph – Autoencoder (AE)

score = log(AE)
⊕ *w* · **grad**

outlier_score
∈ [0, 1]

The issue →





The Solution →

$$\left\{\begin{aligned} \text{ae\_score} &= \frac{\log(1+\text{AE})}{\max(\log(1+\text{AE}))} \\ w &= \exp\left(-\frac{\text{neighbour\_dist}}{1000}\right) \\ v &= 2\cdot\text{ae\_score} + 1\cdot w\cdot\text{rank}(\text{gradient}) \\ \text{outlier\_score} &= \frac{v-\min(v)}{\max(v)-\min(v)} \end{aligned}\right\} \longrightarrow$$

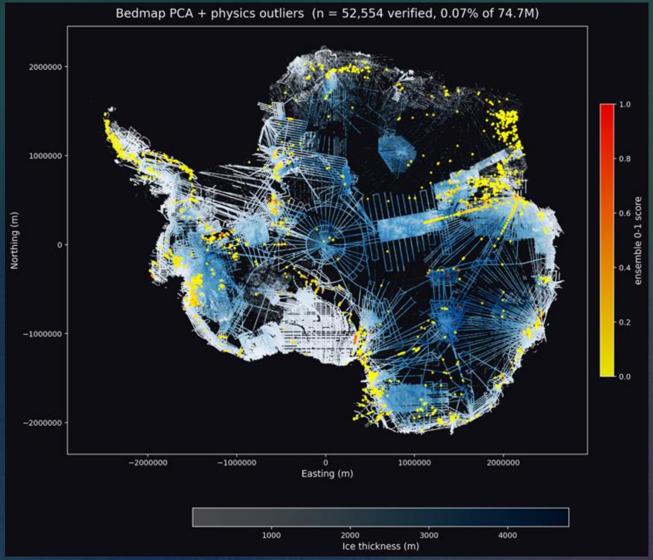
Gives a contiues score

Foto: © ArcticDesire.com Polarreisen, Pexels.com

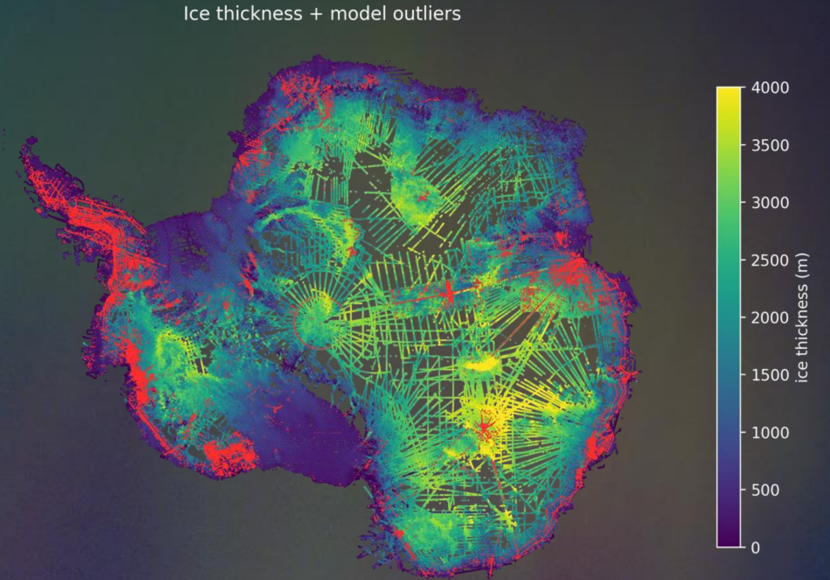
Final Overview

- **All model evaluations,**
  On the ice thickness graph

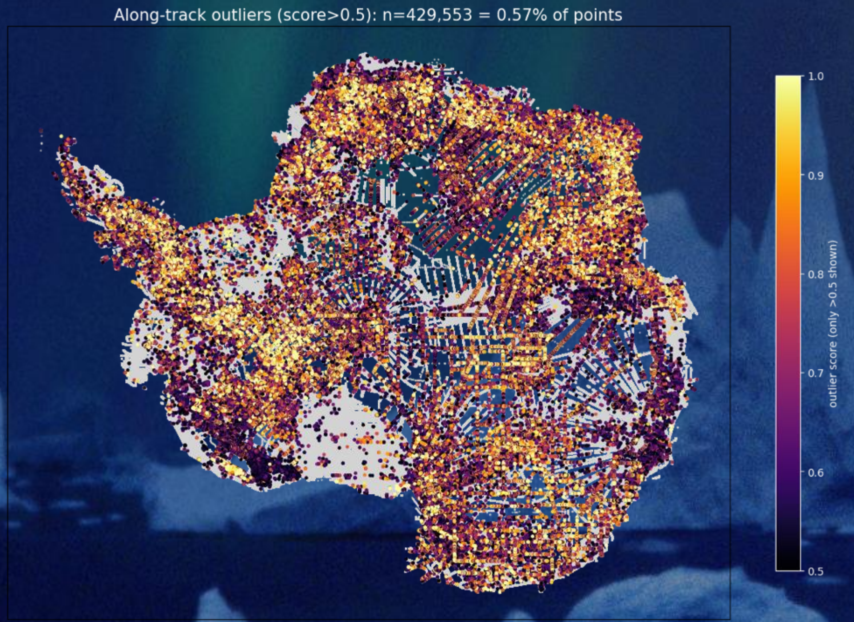
PCA


Bedmap PCA + physics outliers (n = 52,554 verified, 0.07% of 74.7M)

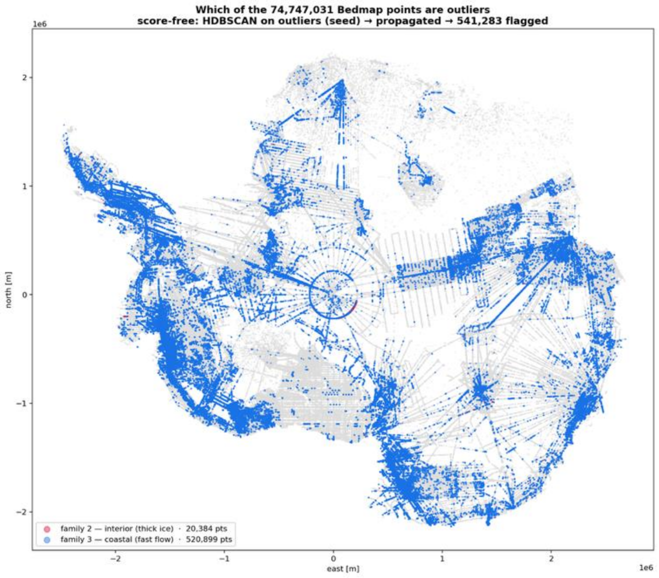
Supervised GNN on Pseudo labels


Ice thickness + model outliers

Along-Track-k-Nearest Neighbors


Along-track outliers (score>0.5): n=429,553 = 0.57% of points

Latent-space Outlier carving


Which of the 74,747,031 Bedmap points are outliers
score-free: HDBSCAN on outliers (seed) → propagated → 541,283 flagged

Graph - Autoencoder

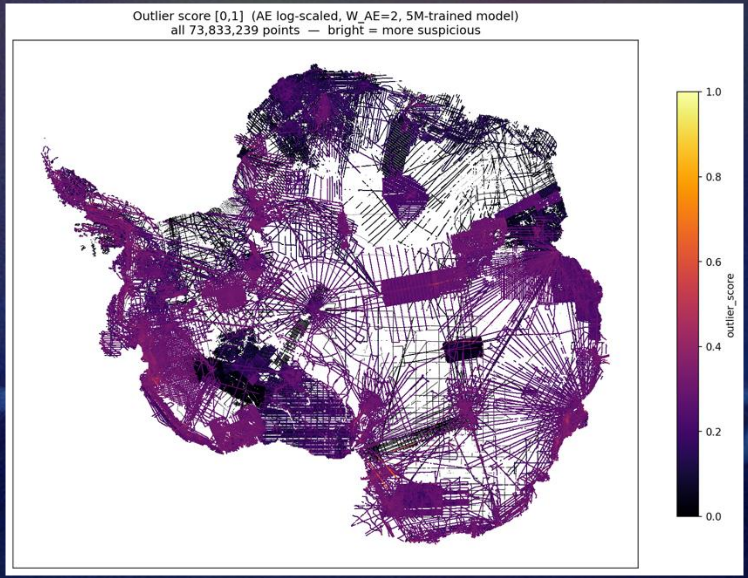

Outlier score [0,1] (AE log-scaled, W_AE=2, 5M-trained model)
all 73,833,239 points — bright = more suspicious

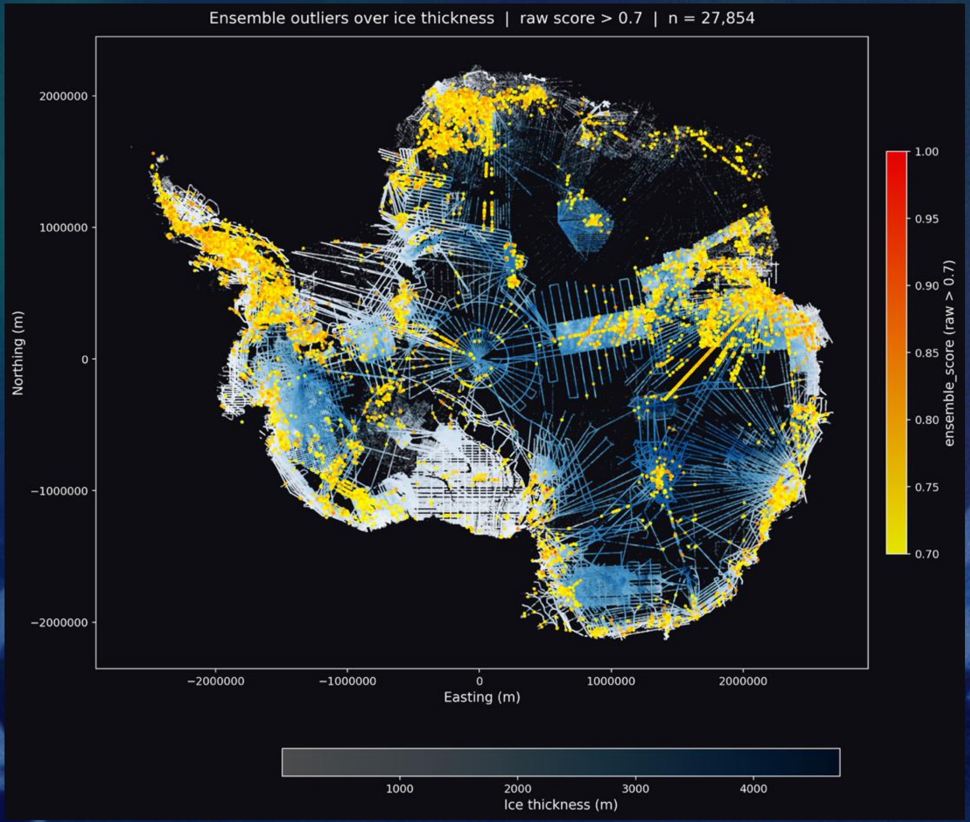
Foto: © Per-Andre Hoffmann

The Ensemble

- **The final ensemble :**

All models in an equally weighted mean.

Numbers of outliers for $p_{out} \geq 0.7$: 27.854



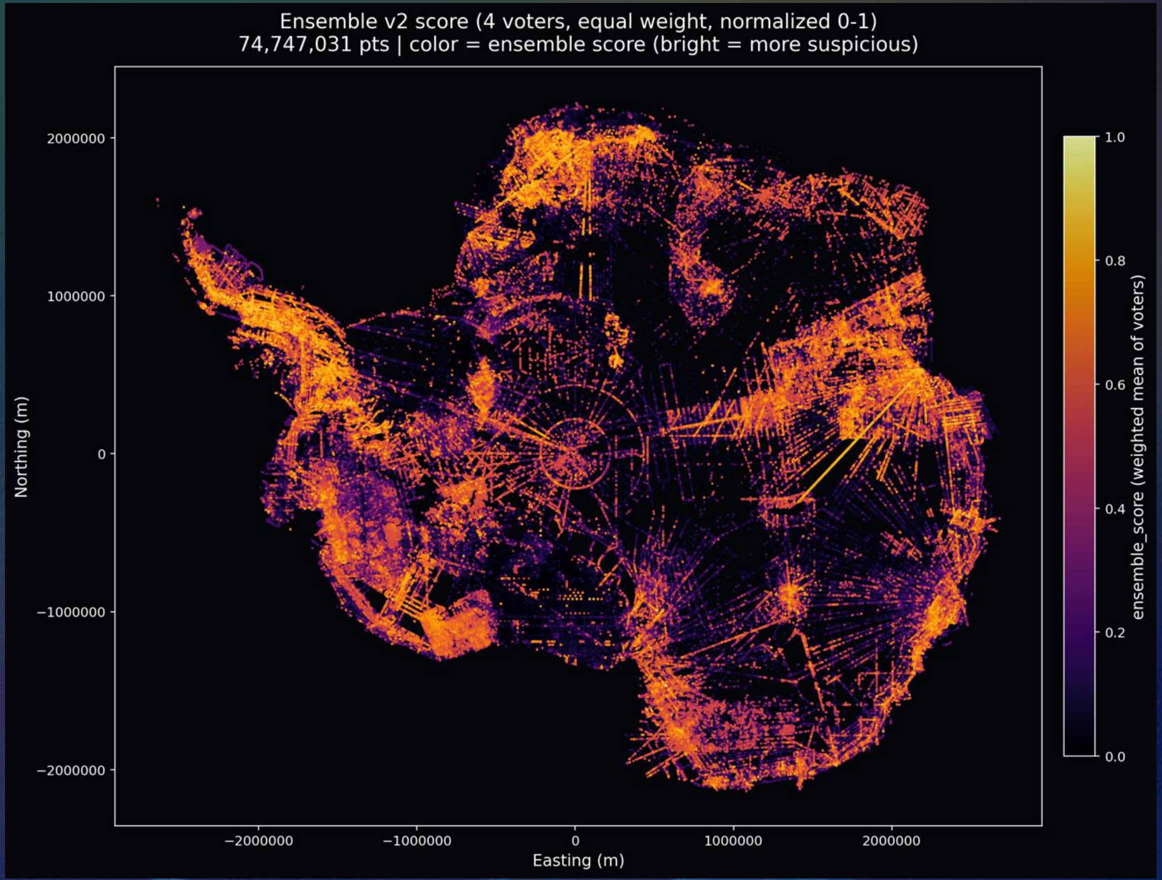


Foto: © Per-Andre Hoffmann

What We'd Do Differently

Suggested improvements

- **For the GNN**

  Pseudo truth labels :

  Consider 25m -> 10m cross-section area for double points

  K-NN-16 :

  Upgrading to 32 nearest neighbours in k-NN – larger local neighbourhood and more context, making it more robust to noise

  Forcing nearest neighbours to have many tracks, e.g. ≥ 5 tracks, for more diversity.

  GNN model :

  3x -> 4x message passing, larger reach for the model to learn from.

  Better Optuna! – only ran for 8 epochs every time.

Foto: © Per-Andre Hoffmann

What We'd Do Differently

Suggested improvements

- **Graph – Autoencoder:**

1. Use velocity vector, not only magnitude
2. Varying hidden layers
3. Better clustering of the feature space

- **Latent Space Outlier Carving:**

1. Find a better training set

- **In general:**

Compare with BedMachine – An accepted model, that approximates Ice-thickness.

Variable analysis, try to find regions in the specific variables that correspond to outliers.

Give individual weights for the final Ensemble model.

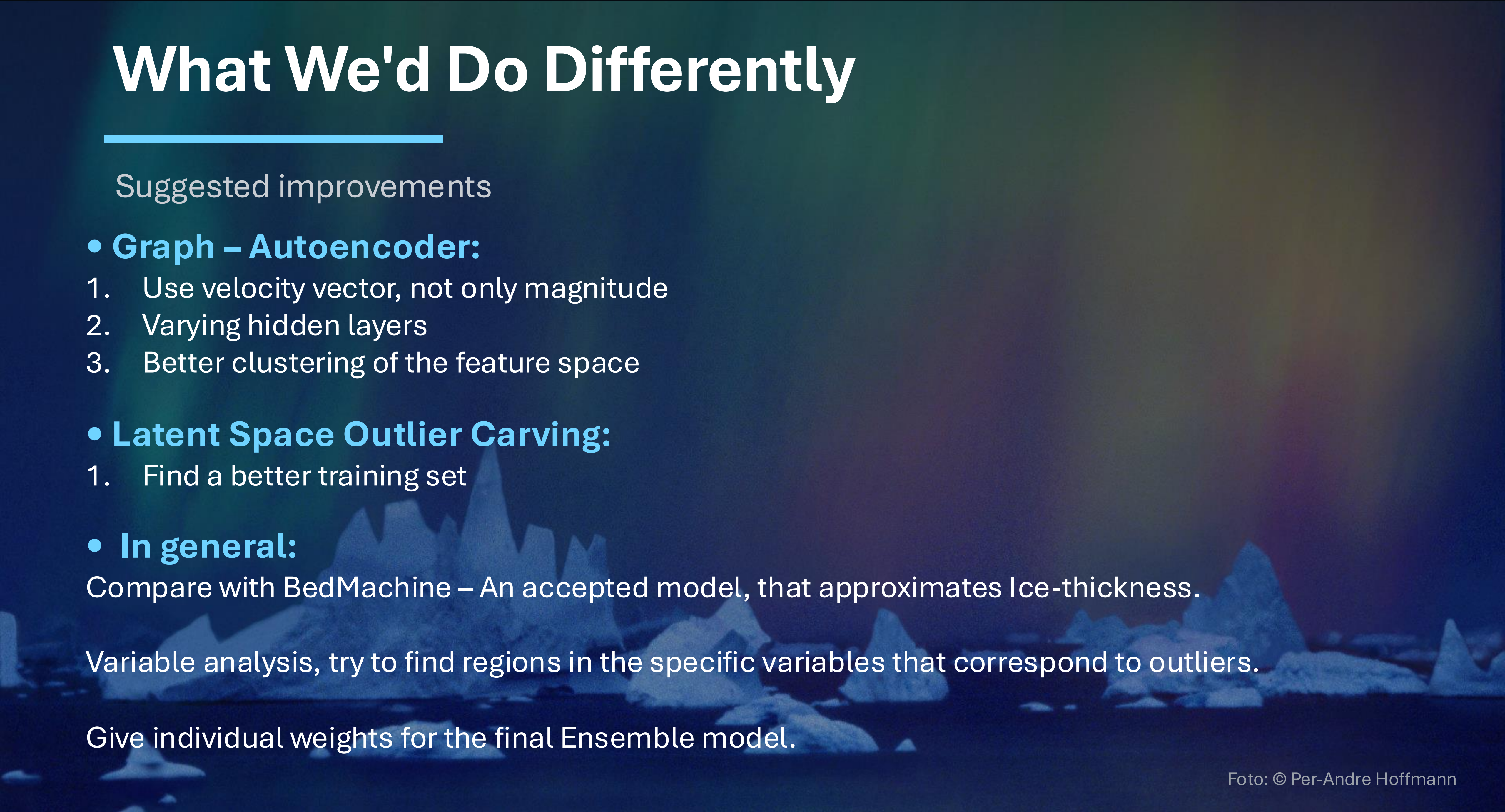
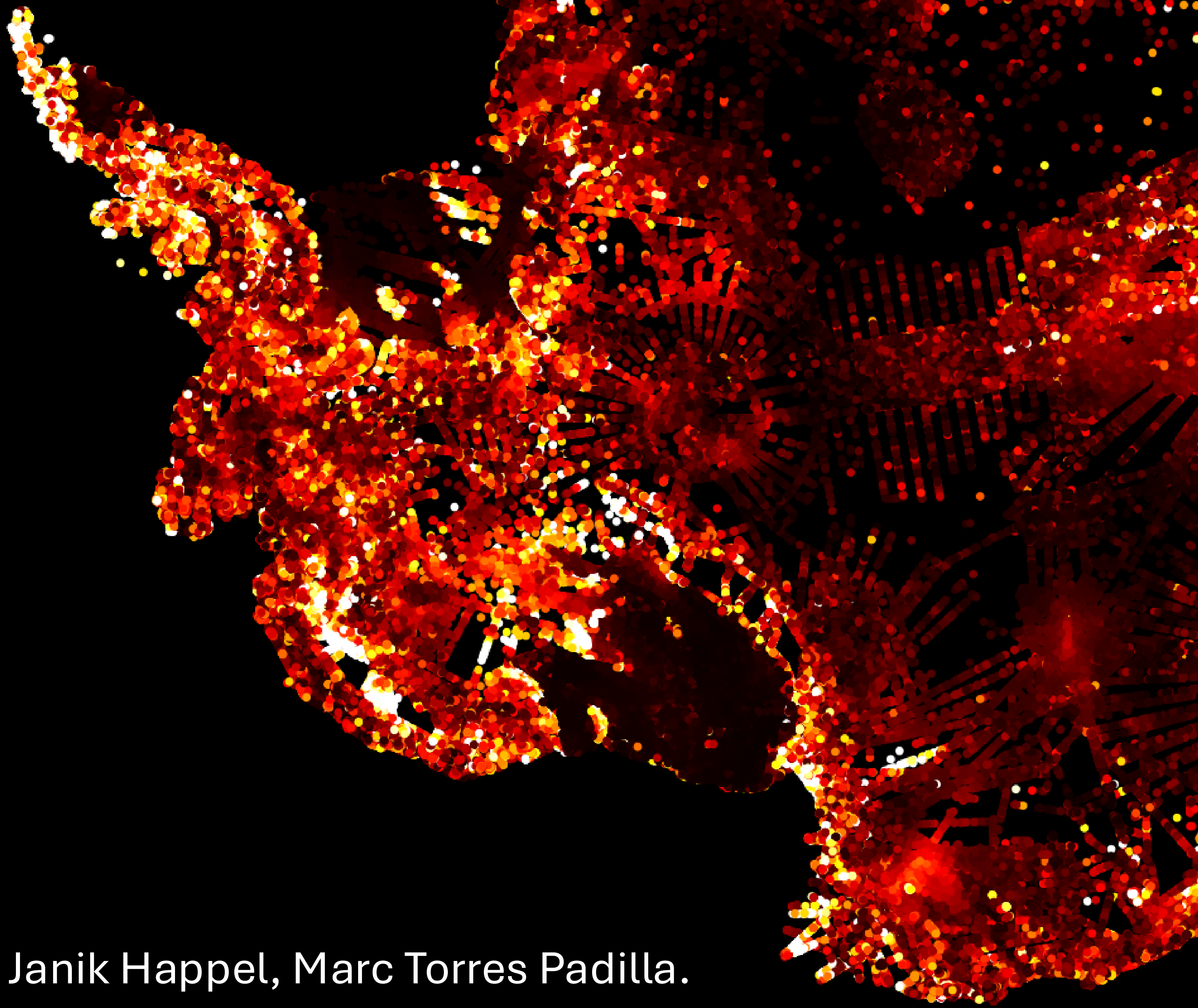


Foto: © Per-Andre Hoffmann

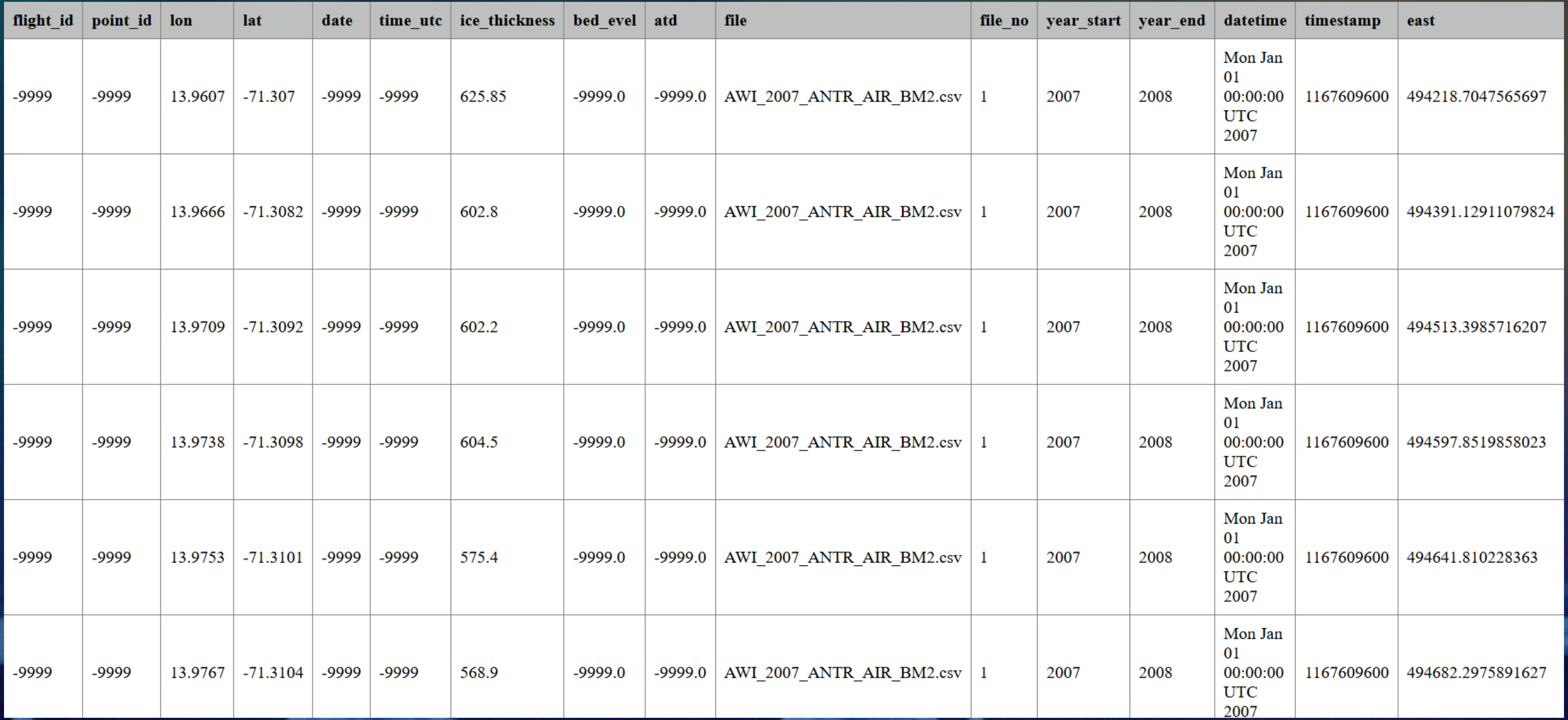


Appendix

Holger Klevang Christiansen, Janik Happel, Marc Torres Padilla.

University of Copenhagen, Applied Machine Learning 2026

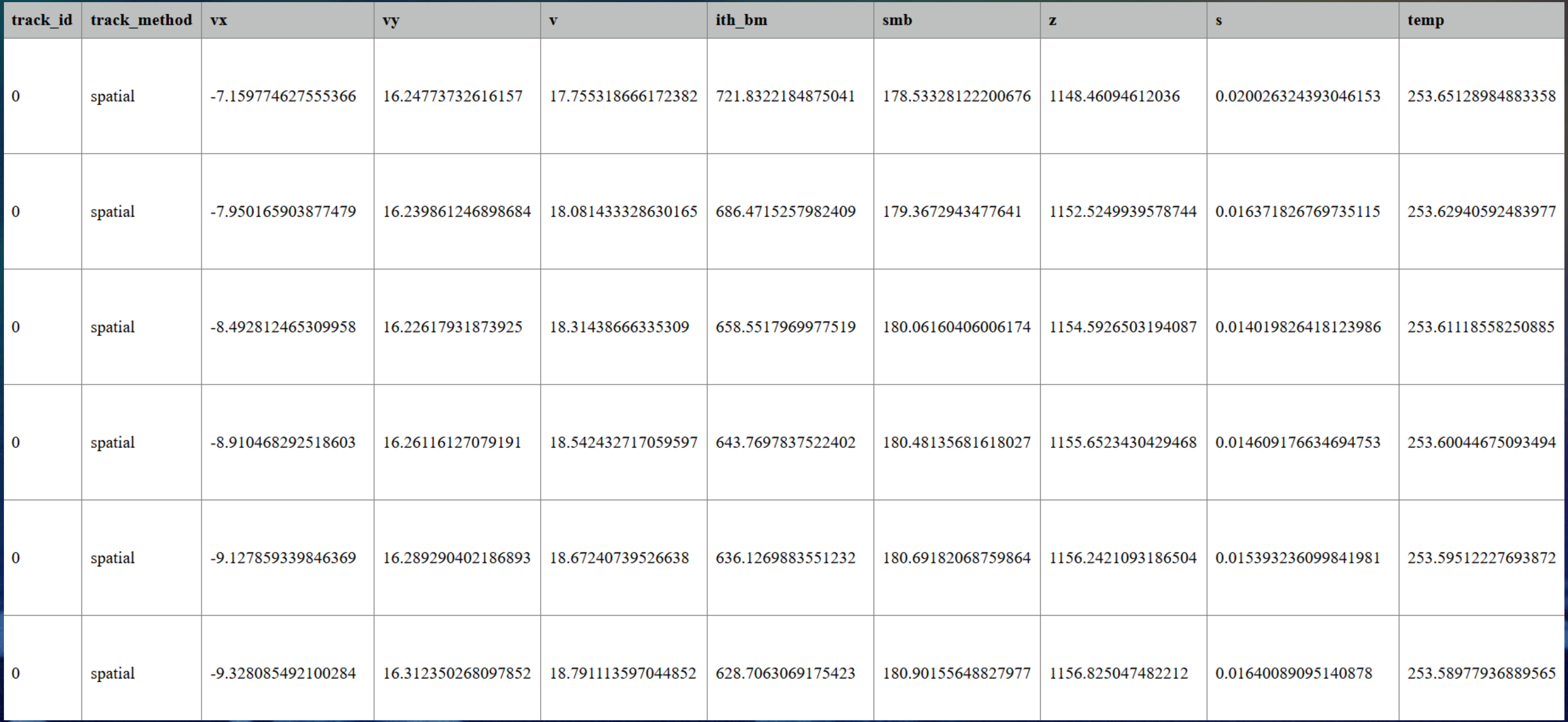
Dataset features 1/2



flight_id	point_id	lon	lat	date	time_utc	ice_thickness	bed_evel	atd	file	file_no	year_start	year_end	datetime	timestamp	east
-9999	-9999	13.9607	-71.307	-9999	-9999	625.85	-9999.0	-9999.0	AWI_2007_ANTR_AIR_BM2.csv	1	2007	2008	Mon Jan 01 00:00:00 UTC 2007	1167609600	494218.7047565697
-9999	-9999	13.9666	-71.3082	-9999	-9999	602.8	-9999.0	-9999.0	AWI_2007_ANTR_AIR_BM2.csv	1	2007	2008	Mon Jan 01 00:00:00 UTC 2007	1167609600	494391.12911079824
-9999	-9999	13.9709	-71.3092	-9999	-9999	602.2	-9999.0	-9999.0	AWI_2007_ANTR_AIR_BM2.csv	1	2007	2008	Mon Jan 01 00:00:00 UTC 2007	1167609600	494513.3985716207
-9999	-9999	13.9738	-71.3098	-9999	-9999	604.5	-9999.0	-9999.0	AWI_2007_ANTR_AIR_BM2.csv	1	2007	2008	Mon Jan 01 00:00:00 UTC 2007	1167609600	494597.8519858023
-9999	-9999	13.9753	-71.3101	-9999	-9999	575.4	-9999.0	-9999.0	AWI_2007_ANTR_AIR_BM2.csv	1	2007	2008	Mon Jan 01 00:00:00 UTC 2007	1167609600	494641.810228363
-9999	-9999	13.9767	-71.3104	-9999	-9999	568.9	-9999.0	-9999.0	AWI_2007_ANTR_AIR_BM2.csv	1	2007	2008	Mon Jan 01 00:00:00 UTC 2007	1167609600	494682.2975891627

Foto: © Per-Andre Hoffmann

Dataset features 2/2



track_id	track_method	vx	vy	v	ith_bm	smb	z	s	temp
0	spatial	-7.159774627555366	16.24773732616157	17.755318666172382	721.8322184875041	178.53328122200676	1148.46094612036	0.020026324393046153	253.65128984883358
0	spatial	-7.950165903877479	16.239861246898684	18.081433328630165	686.4715257982409	179.3672943477641	1152.5249939578744	0.016371826769735115	253.62940592483977
0	spatial	-8.492812465309958	16.22617931873925	18.31438666335309	658.5517969977519	180.06160406006174	1154.5926503194087	0.014019826418123986	253.61118558250885
0	spatial	-8.910468292518603	16.26116127079191	18.542432717059597	643.7697837522402	180.48135681618027	1155.6523430429468	0.014609176634694753	253.60044675093494
0	spatial	-9.127859339846369	16.289290402186893	18.67240739526638	636.1269883551232	180.69182068759864	1156.2421093186504	0.015393236099841981	253.59512227693872
0	spatial	-9.328085492100284	16.312350268097852	18.791113597044852	628.7063069175423	180.90155648827977	1156.825047482212	0.01640089095140878	253.58977936889565

Foto: © Per-Andre Hoffmann

PCA

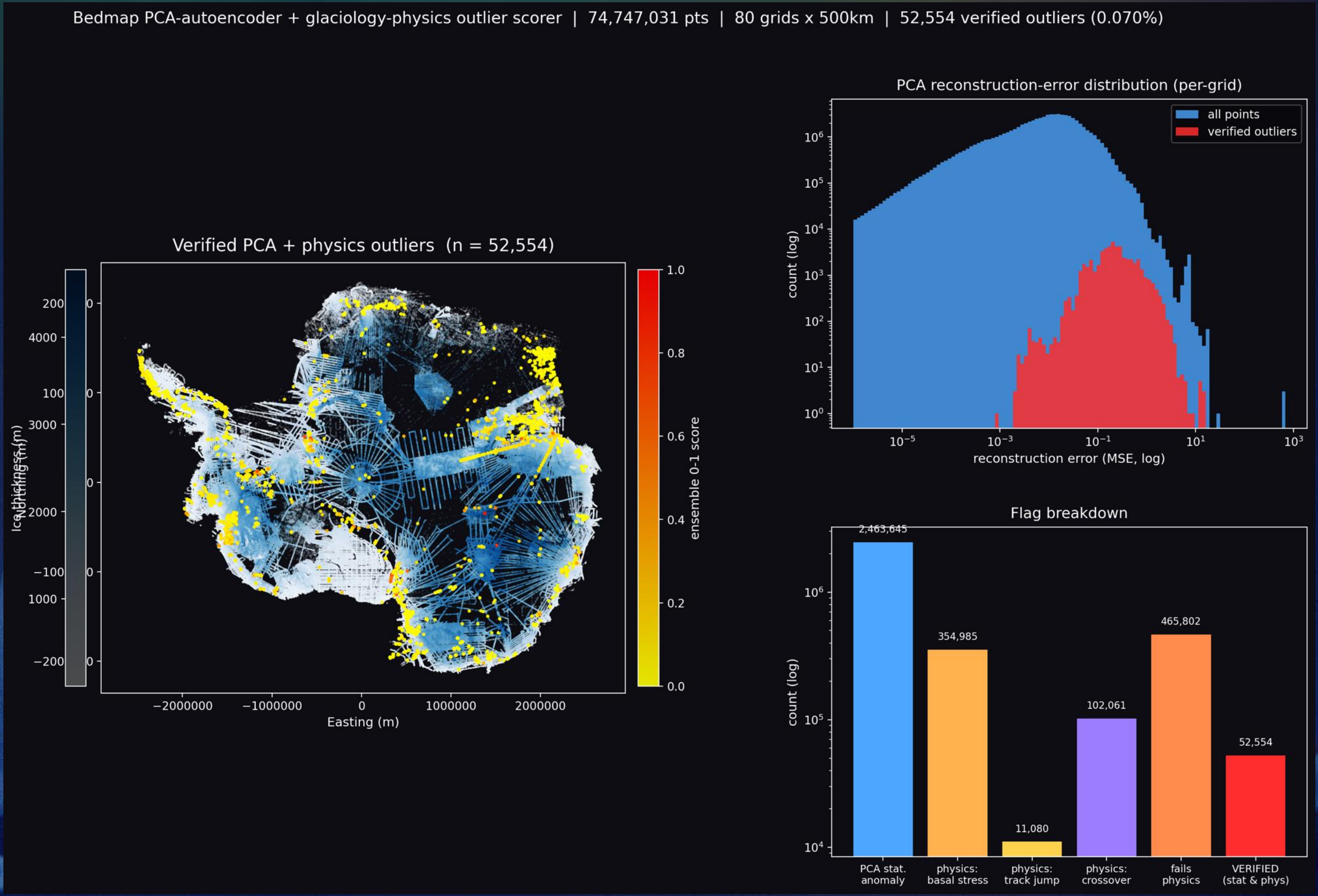


Foto: © Per-Andre Hoffmann

Physics-Backed PCA

- **Why use PCA?** — Strict and linear algorithm
- **Geographic grids** — To avoid the difference between costal and inland effects

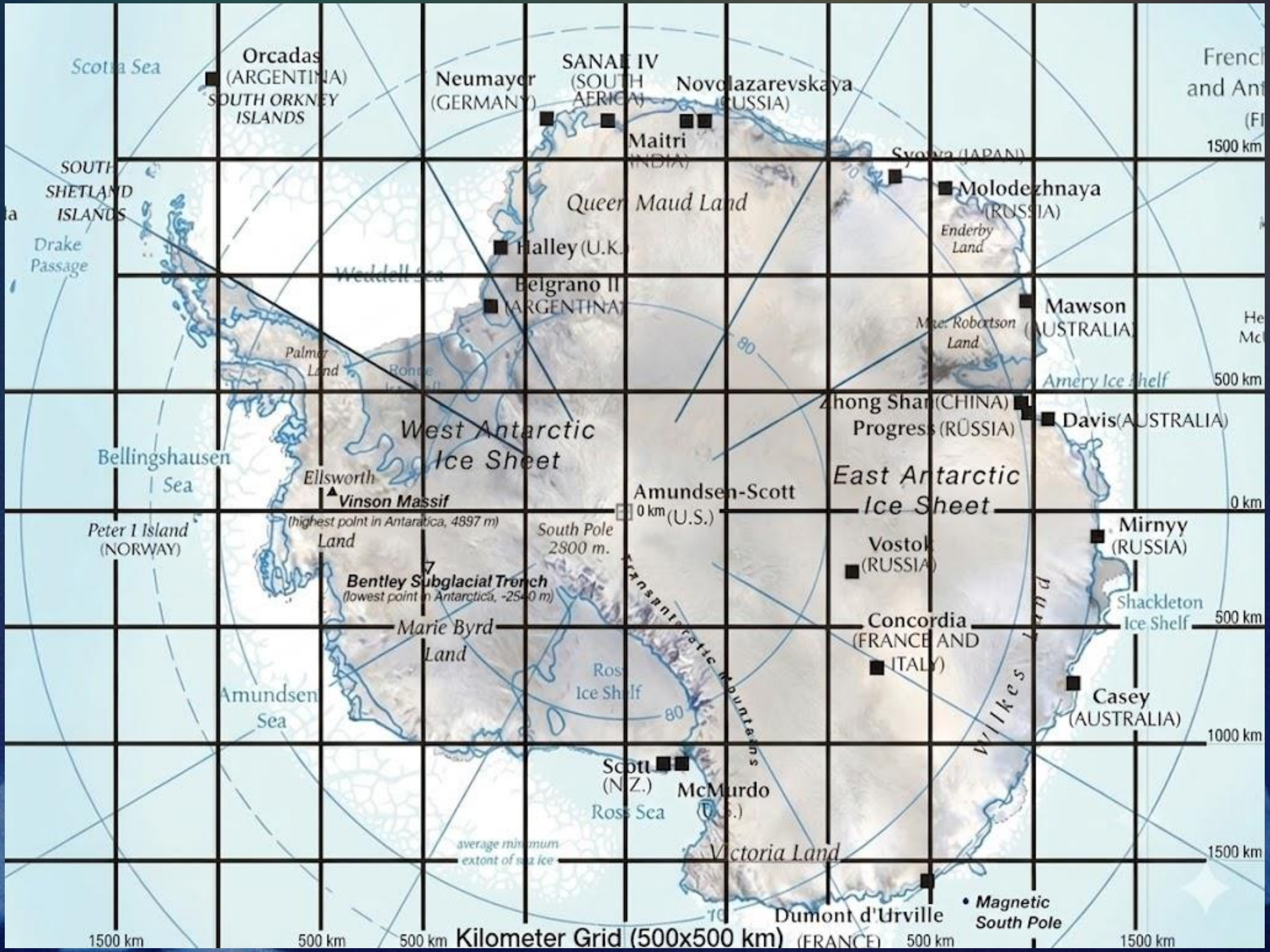
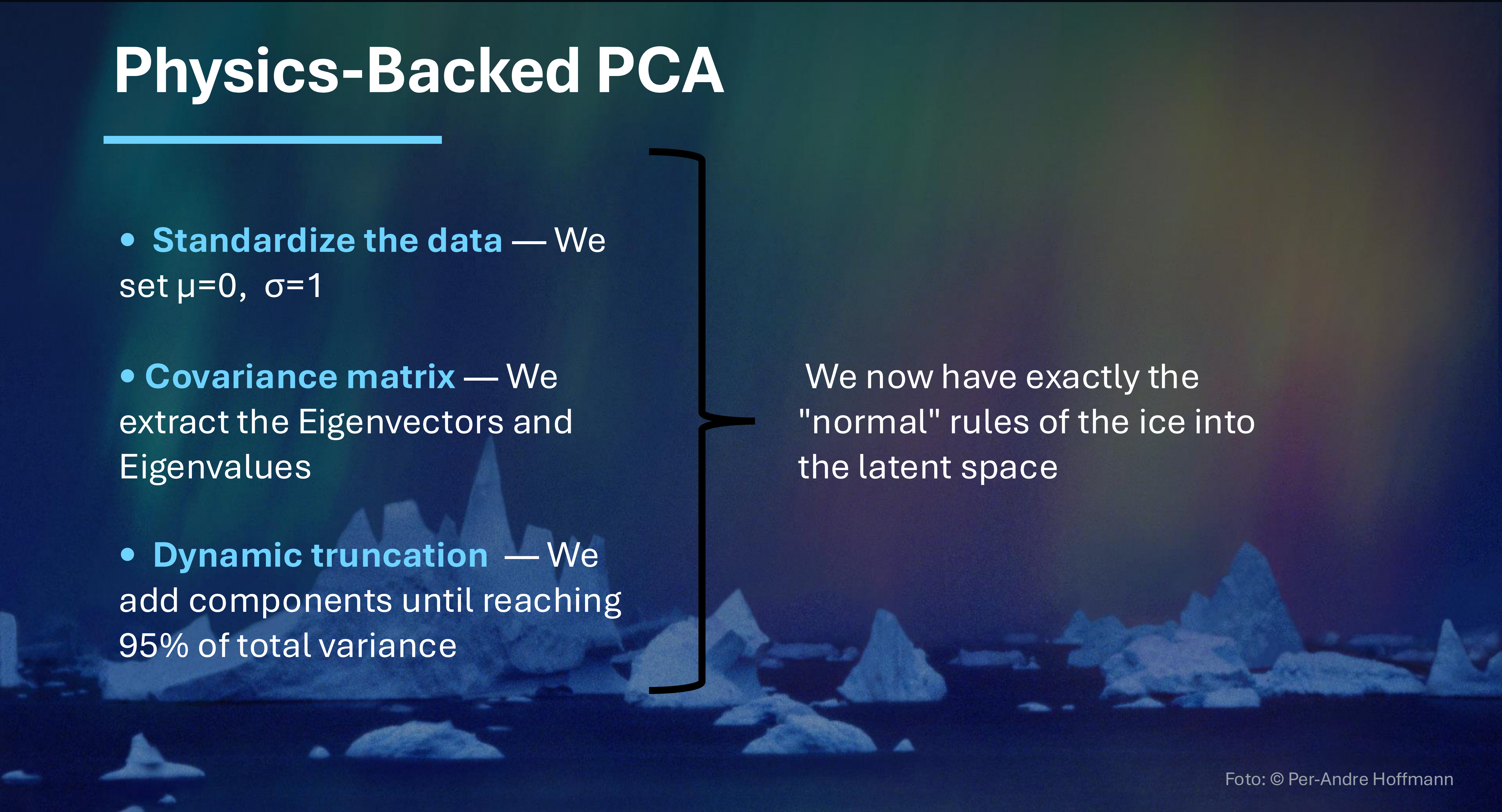


Foto: © Per-Andre Hoffmann

Physics-Backed PCA



- **Standardize the data** — We set μ=0, σ=1

- **Covariance matrix** — We extract the Eigenvectors and Eigenvalues

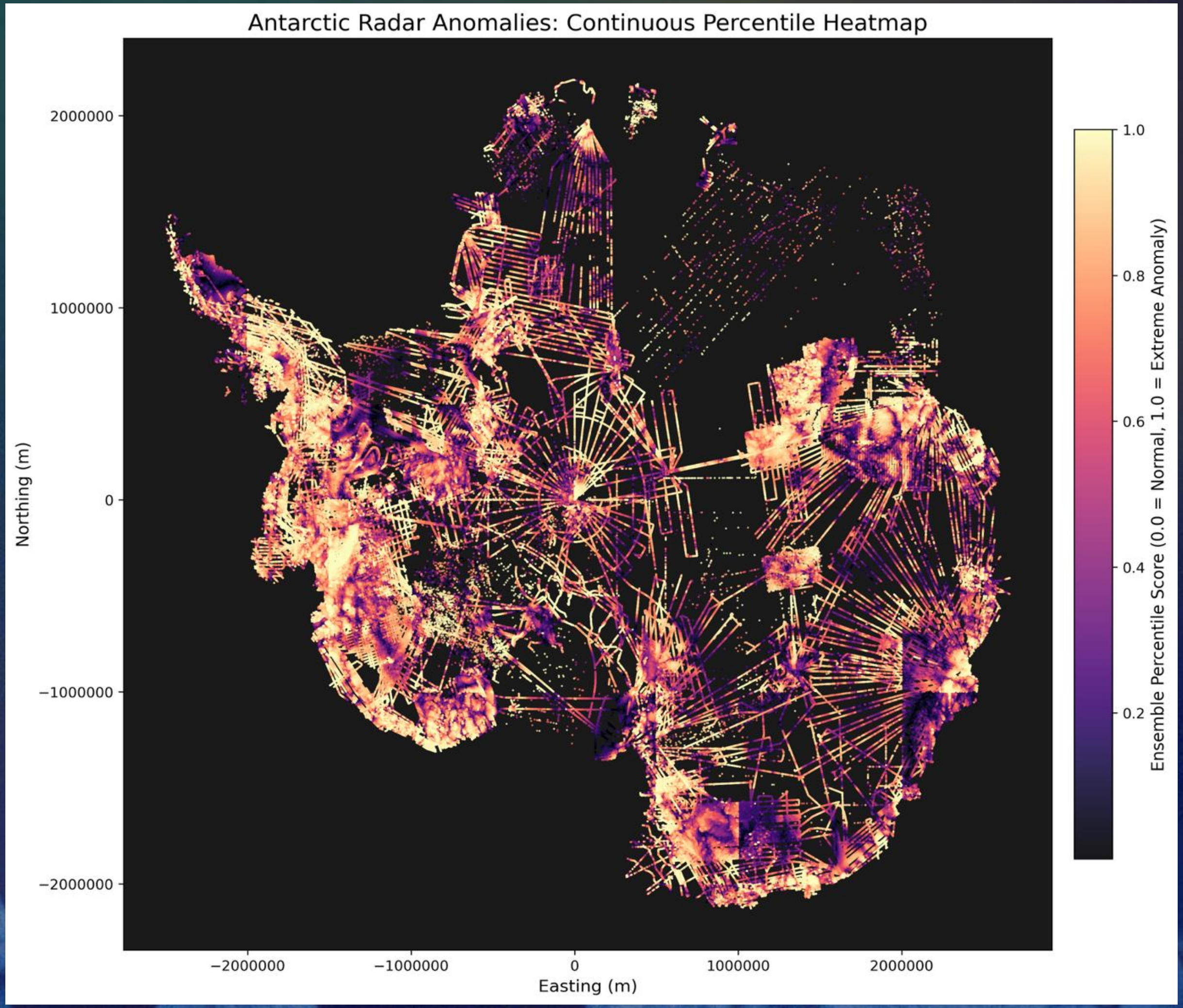
- **Dynamic truncation** — We add components until reaching 95% of total variance

We now have exactly the "normal" rules of the ice into the latent space

Foto: © Per-Andre Hoffmann

Physics-Backed PCA

- **Latent Reconstruction** — The data is projected into the truncated latent space and then back to the latent space
- **Reconstruction error** — Using mean squared error across all features
- **MAD** — Standard deviation explodes. MAD remains stable even if half the grid is corrupted
- **Min-Max scaler** — To classify the points between 0 and 1, and avoid infinite outliers ruining everything



Using only 2.000.000 points

Foto: © Per-Andre Hoffmann

Physics-Backed PCA

We can fix it!

- **Glaciological Physics Engine** — We pass now the results through a glaciological laws corrector

- **Along-Track Variance** — With the rate of change, we check for massive jumps on the same flight
- **Spatio-Temporal Intersection Validation** — Check the cross section for two different flights.

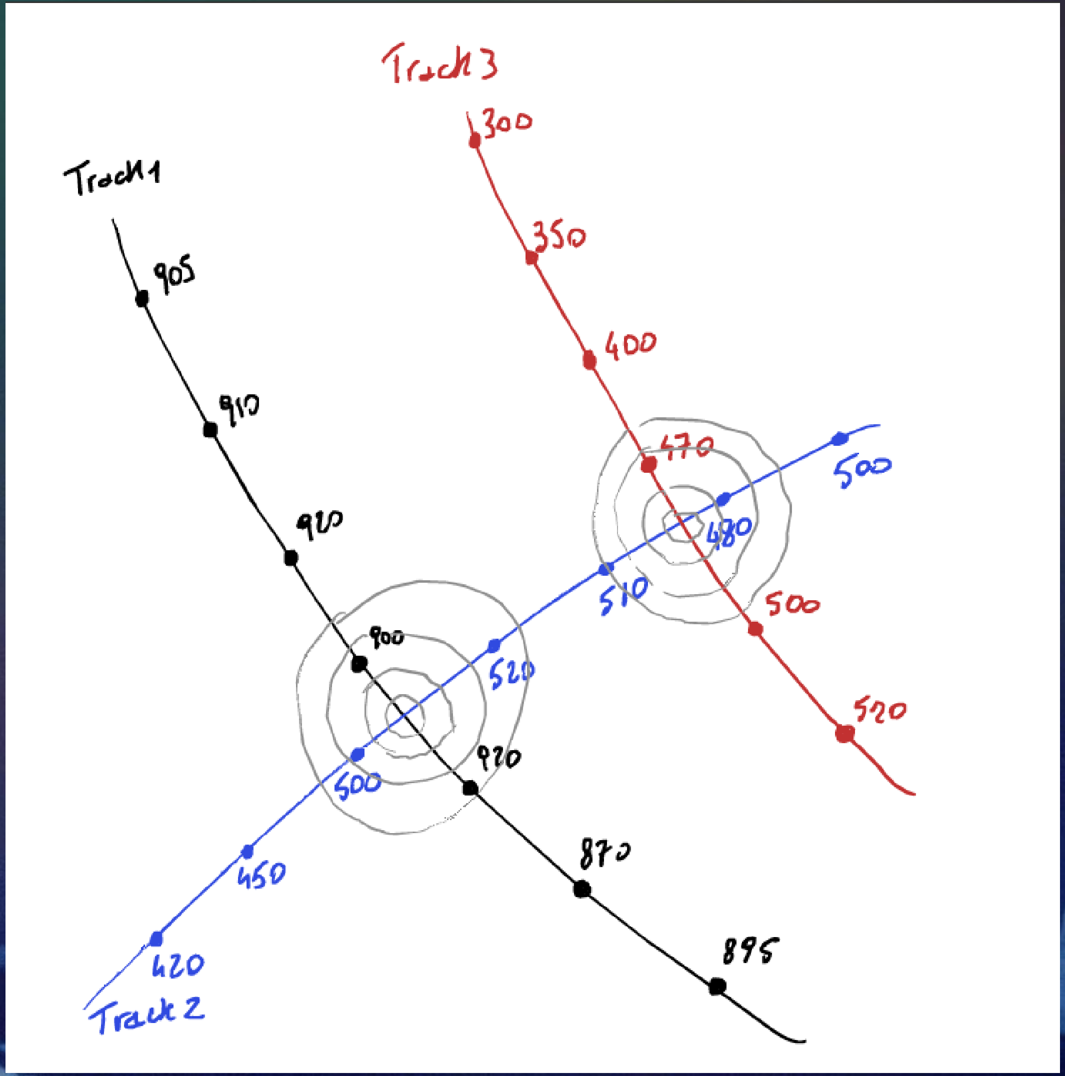


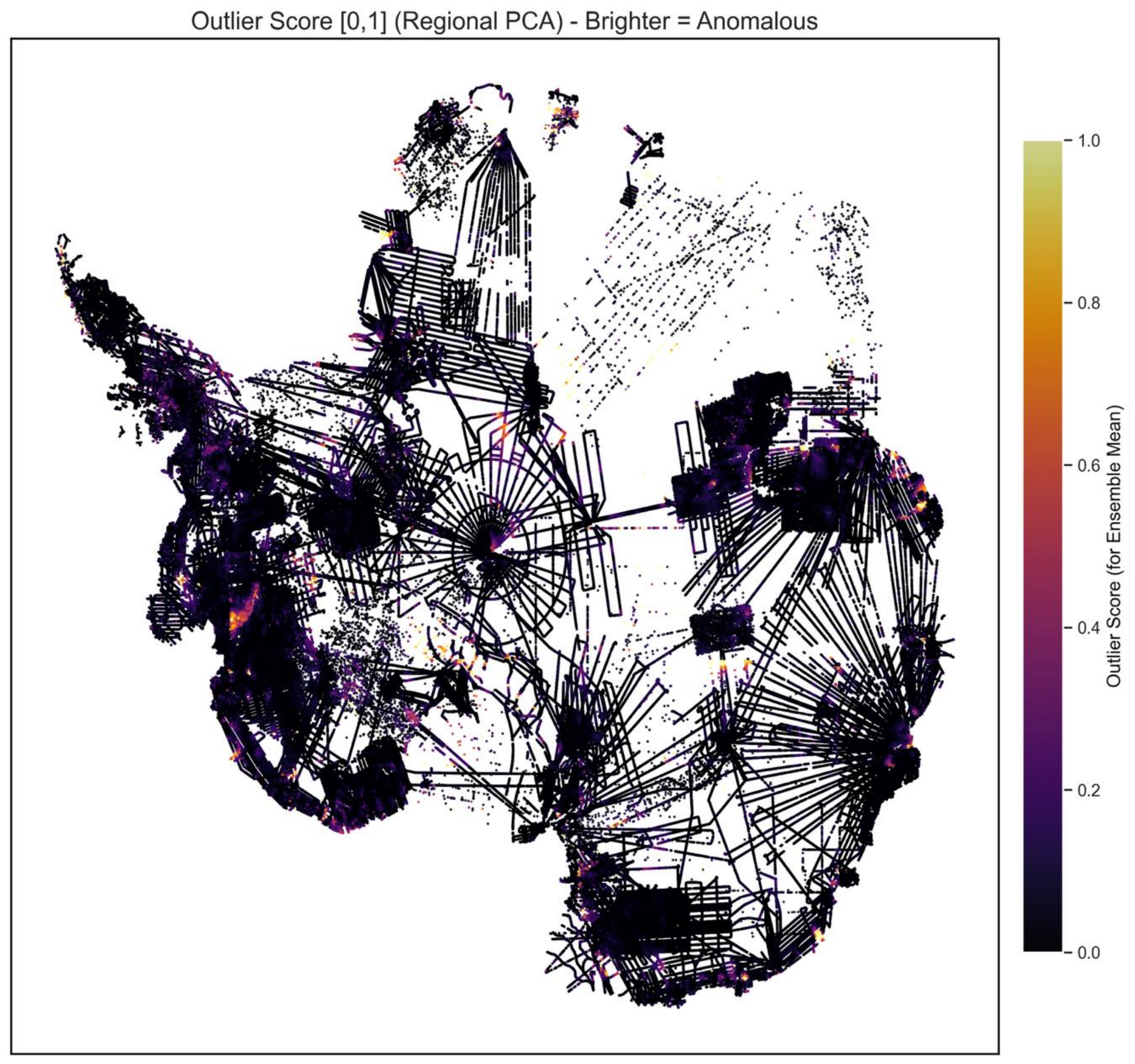
Foto: © Per-Andre Hoffmann

Physics-Backed PCA

Almost there!

- **Basal Shear Stress** — Using ice density, gravity, ice thickness, and surface slope

- **Epistemic Uncertainty & Temporal Weighting** — Prioritize modern data when considering crossovers

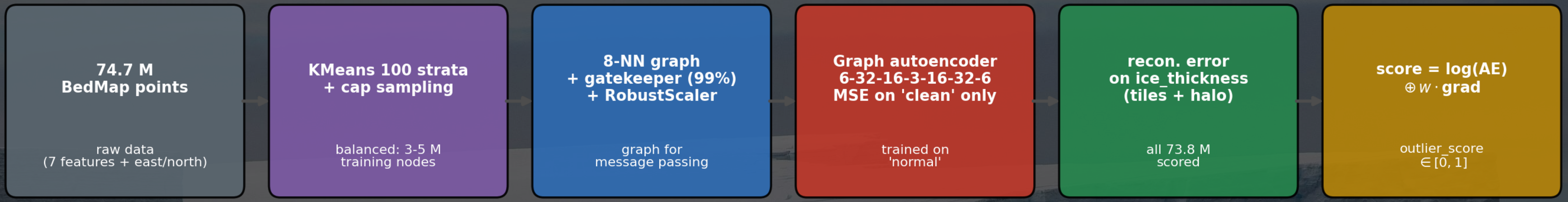


Using only 2.000.000 points

Foto: © Per-Andre Hoffmann

Graph – Autoencoder (AE)

autoencoder output: outlier_score $\in [0, 1]$ ***per point***



74.7 M BedMap points
raw data
(7 features + east/north)

KMeans 100 strata + cap sampling
balanced: 3-5 M
training nodes

8-NN graph + gatekeeper (99%) + RobustScaler
graph for
message passing

Graph autoencoder 6-32-16-3-16-32-6 MSE on 'clean' only
trained on
'normal'

recon. error on ice_thickness (tiles + halo)
all 73.8 M
scored

score = log(AE) $\oplus\, w \cdot$ **grad**
outlier_score
$\in [0, 1]$

<- the Graph autoencoder sits here ->

Idea: same autoencoder, but each point also sees its neighbours

How:

build an **8-NN graph** in (east, north)
Linear → SAGEConv: $h_i \leftarrow W_\ell \cdot h_i + W_r \cdot \mathrm{mean}(\text{neighbours})$

Fixes the issues:

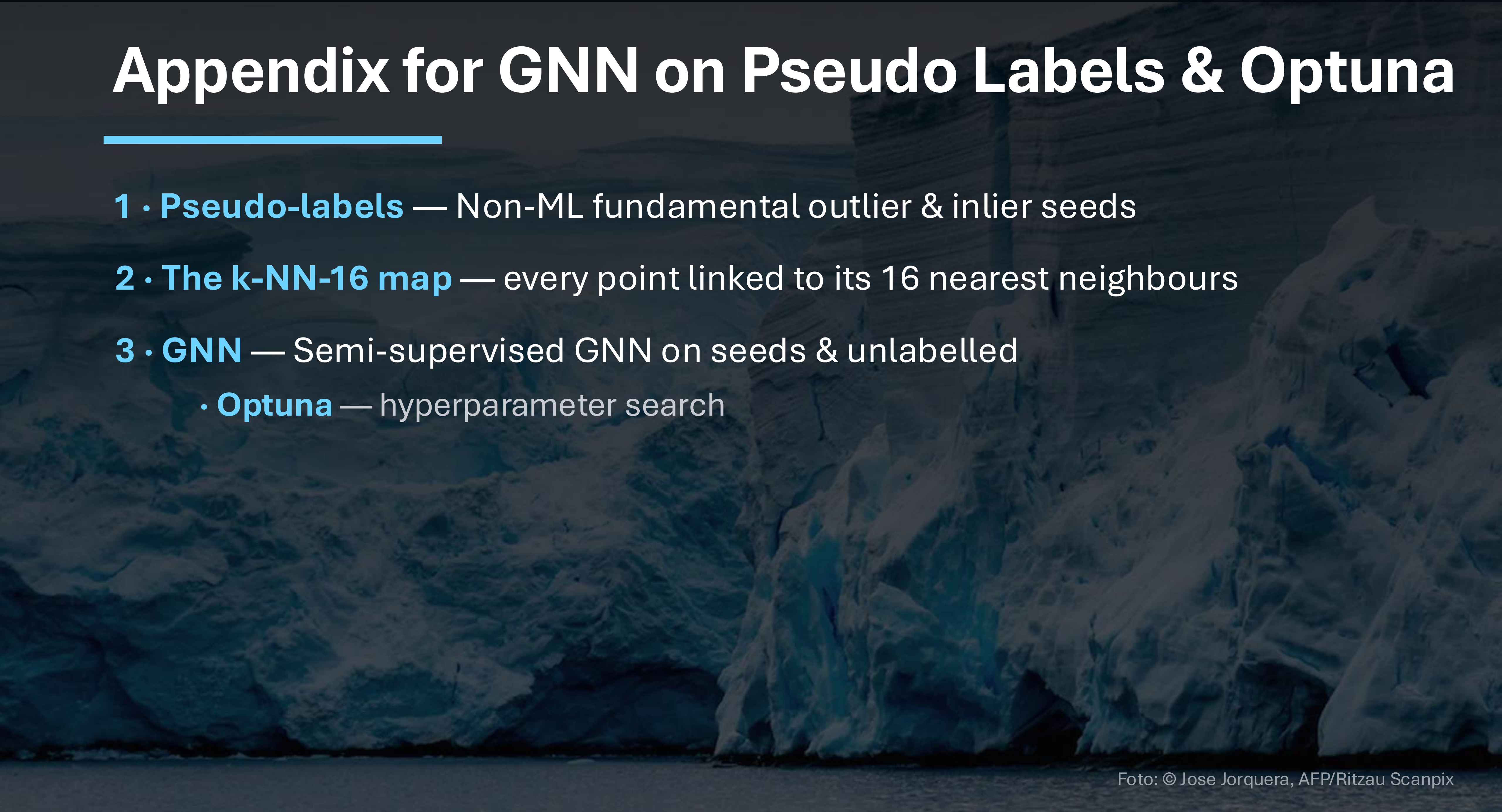
spatial context ✓ — neighbours feed every layer
reconstruction = **local** normal, not global mean ✓
rare-but-real → handled later (balanced training + gradient check)

Architecture: 6 → 32 → 16 → 3 → 16 → 32 → 6 (SAGEConv, latent = 3)

Outlier = high reconstruction error on ice_thickness

Foto: © ArcticDesire.com Polarreisen, Pexels.com

Appendix for GNN on Pseudo Labels & Optuna



1 · **Pseudo-labels** — Non-ML fundamental outlier & inlier seeds

2 · **The k-NN-16 map** — every point linked to its 16 nearest neighbours

3 · **GNN** — Semi-supervised GNN on seeds & unlabelled
  - **Optuna** — hyperparameter search

Foto: © Jose Jorquera, AFP/Ritzau Scanpix

A.1.1 · Candidates for pseudo-labels

Non-ML, pure geometry motivated selection

- **Double hit** — 2 tracks within 25 m, i.e. same spot measured twice (occurs 21.7 million times)
- **Disagree** → outlier; agree → inlier
- **Deciding factor?** :
  Local support area (2km radius)
  more on next slides!

A crossover measured twice, independently

disagree → one is wrong → OUTLIER

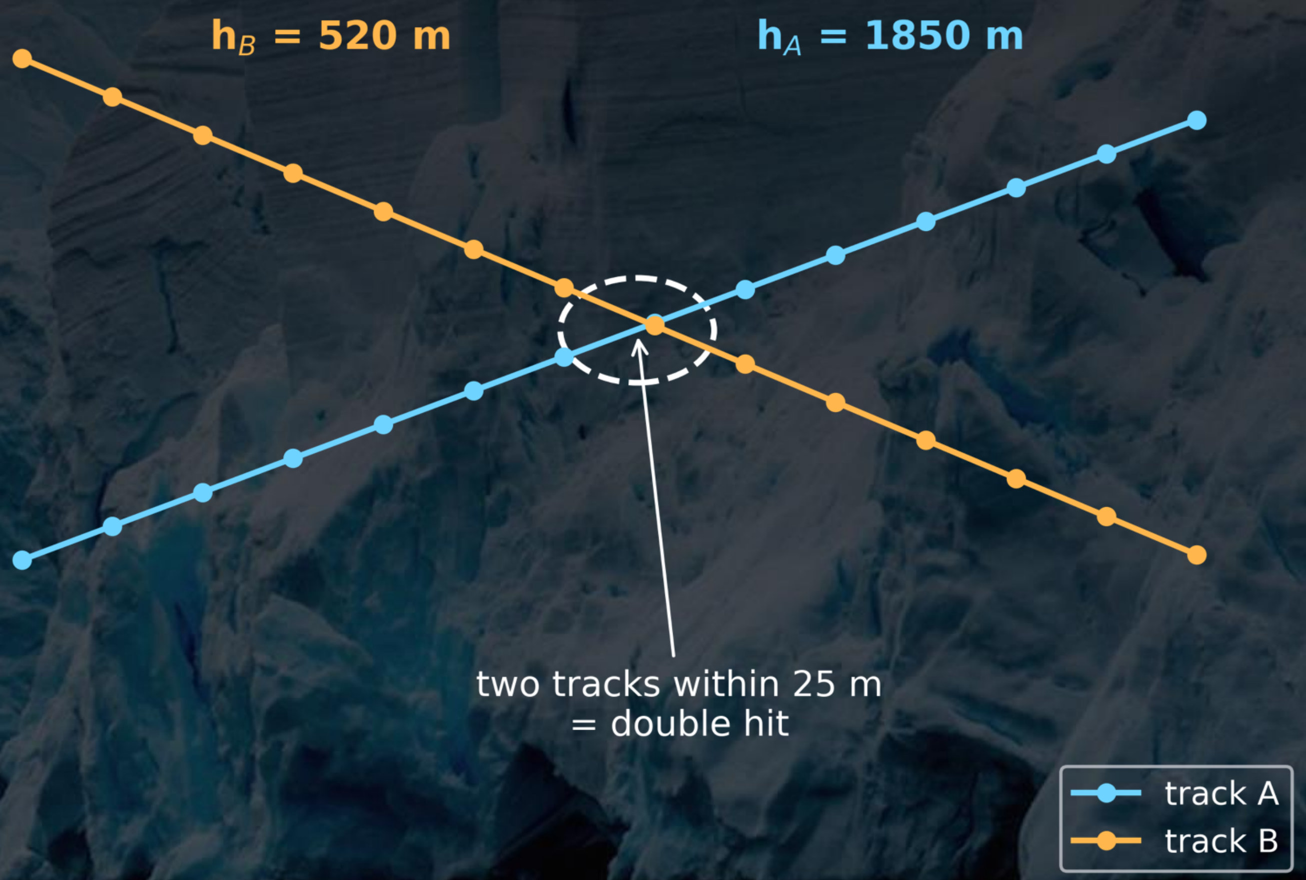
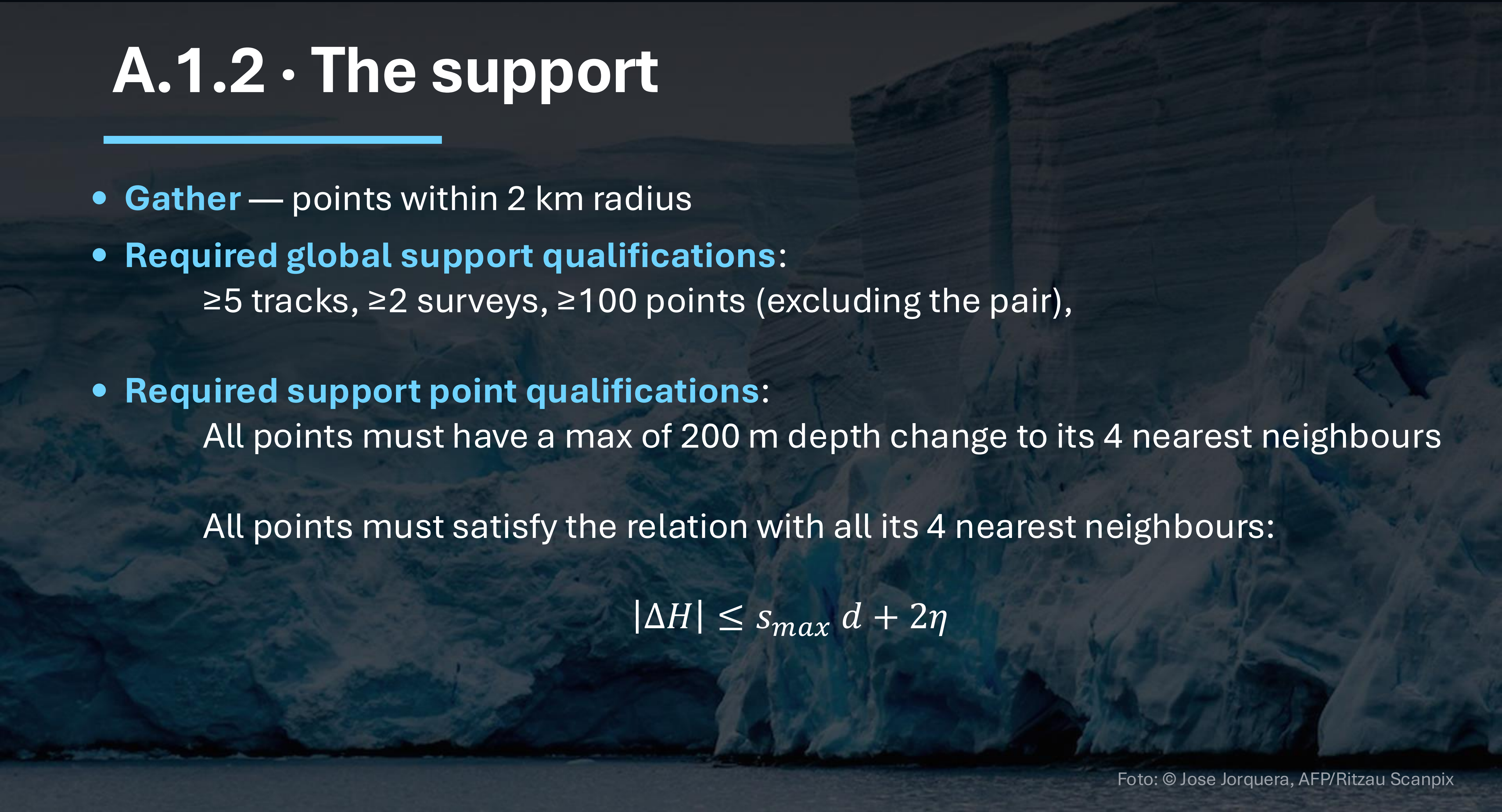


Foto: © Jose Jorquera, AFP/Ritzau Scanpix



A.1.2 · The support

- **Gather** — points within 2 km radius
- **Required global support qualifications**:
  ≥5 tracks, ≥2 surveys, ≥100 points (excluding the pair),
- **Required support point qualifications**:
  All points must have a max of 200 m depth change to its 4 nearest neighbours

  All points must satisfy the relation with all its 4 nearest neighbours:

$$|\Delta H| \leq s_{max}\, d + 2\eta$$

Foto: © Jose Jorquera, AFP/Ritzau Scanpix

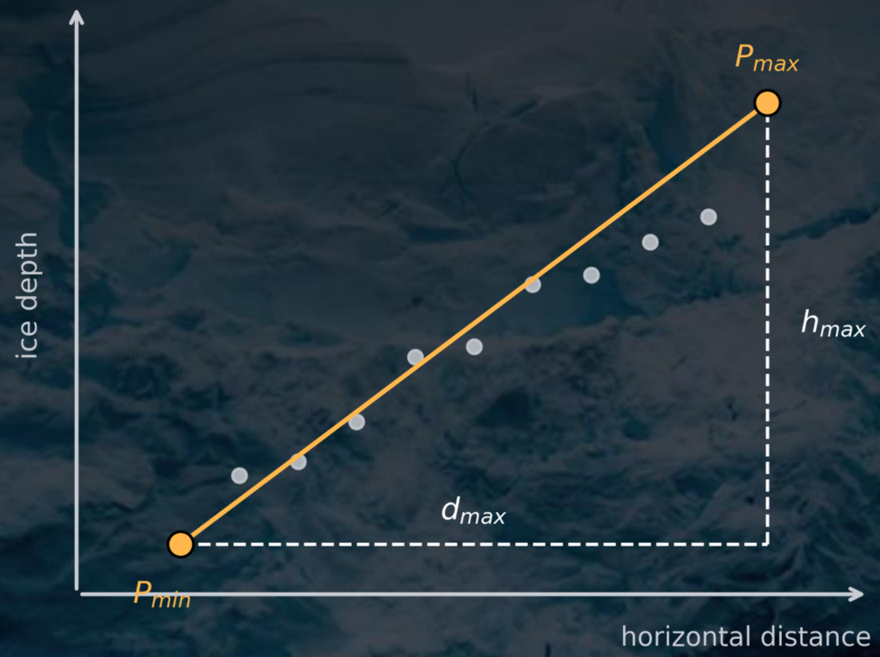
A.1.2 · The support

remove all points that don't satisfy this and the 200m ice depth criterion. If below <100 points or more then 30% of the total support has been removed, discard the pair.

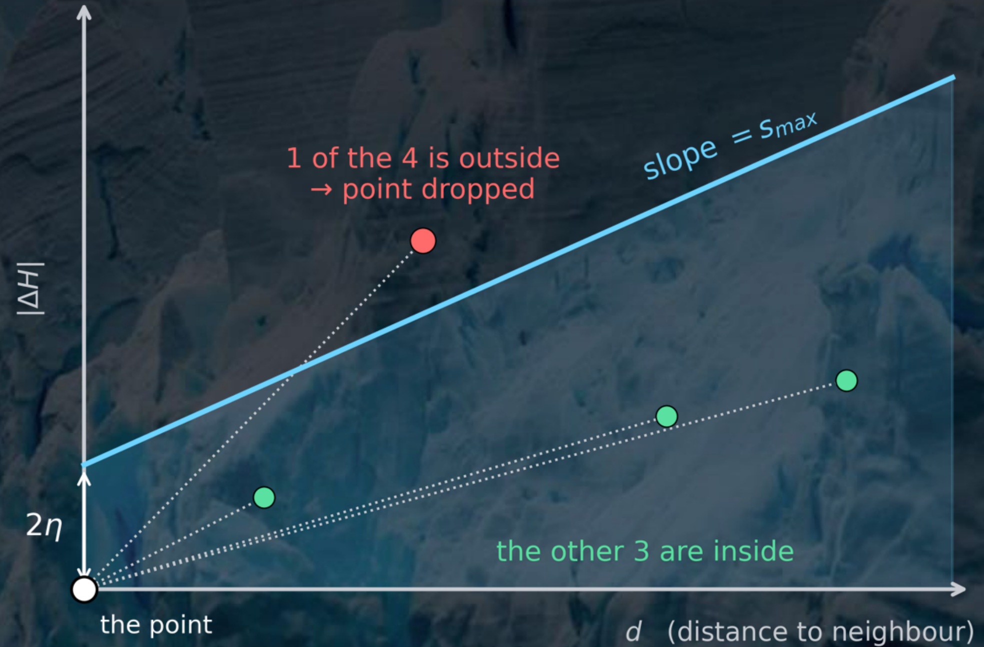
$$|\Delta H| \leq s_{max}\, d + 2\eta$$

1 · the support's steepest slope

$s_{max} = 2\, h_{max}/d_{max}$



2 · a point vs its 4 nearest neighbours



$|\Delta H|$ = ice-depth difference to a neighbour · d = distance to it · $\eta = \max(15\,\text{m},\ 0.01 \times$ support ice depth$)$

Foto: © Jose Jorquera, AFP/Ritzau Scanpix

A.1.3 · The cone verdict

Step 2 — judged vs OTHER tracks

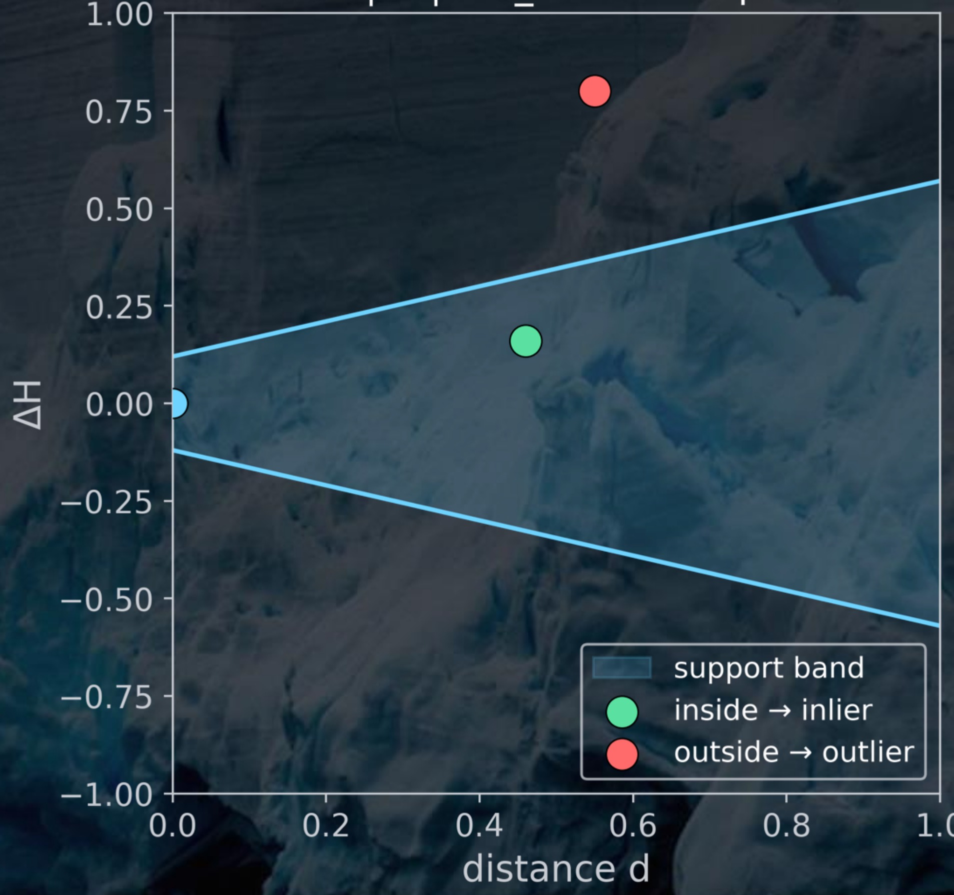
- **An inlier needs:**

- Support in all 8 octants
- To be inside all bands

- **An outlier needs:**

- Support in all 8 octants
- 1 missing band or more
- Its double hit partner to pass

surrounded?
(≥1 support in all 8 octants)



within the band?
$|\Delta H| \leq s\_max \cdot d + 2\eta$

Foto: © Jose Jorquera, AFP/Ritzau Scanpix

A.1.4 · The resulting pseudo-labels

- **Overview** — of the processed pseudo-labels:

21.7 million pairs

- **9.9 million (discarded)**
    - Pairs that didn’t pass the support check
    - Double outliers
    - Double disagreeing inliers $(|\Delta H| > 2s_{max}\, d + 2\eta)$

↓

11.1 million double inliers + 0.7 million inlier/outlier pairs

↓

646k unique inliers + 39k unique outliers

5k conflicted points before these numbers were discarded (labelled both as inlier and outlier)

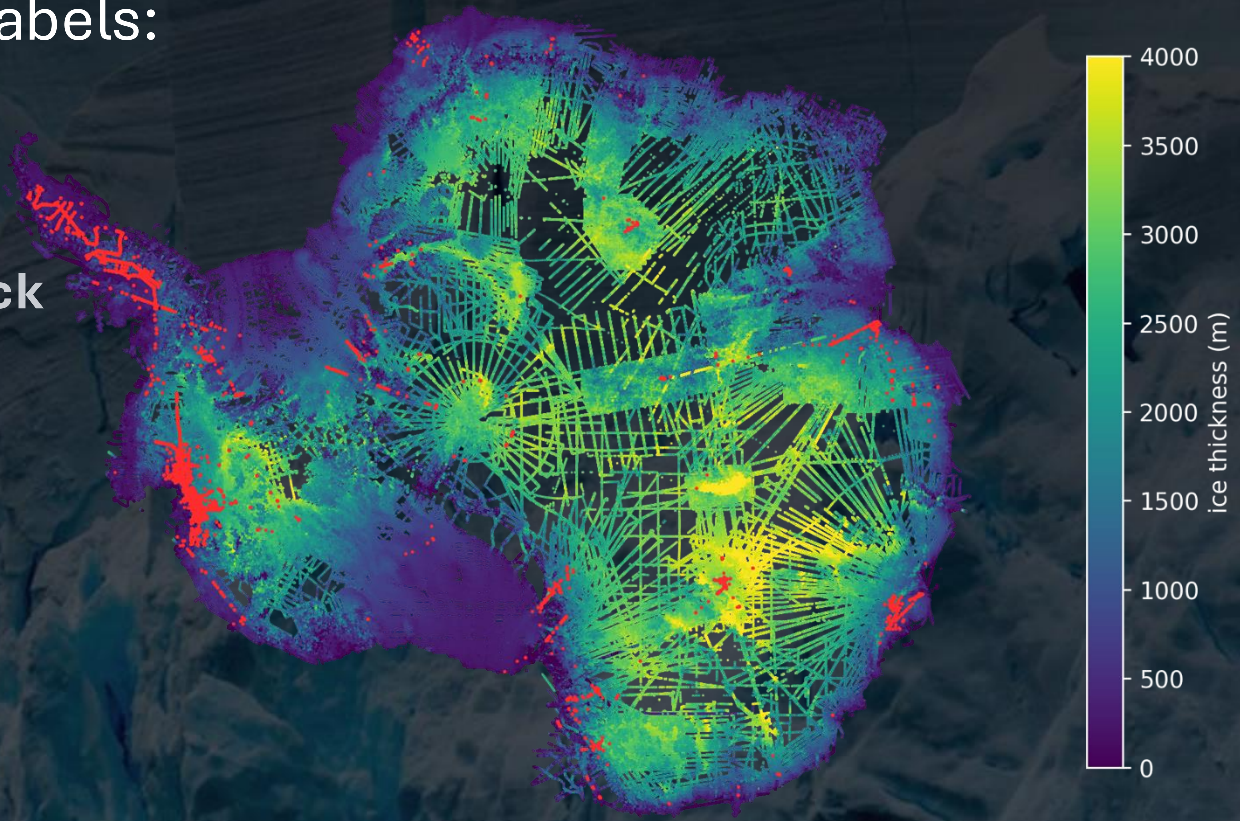

Ice thickness + outlier seeds

Foto: © Jose Jorquera, AFP/Ritzau Scanpix

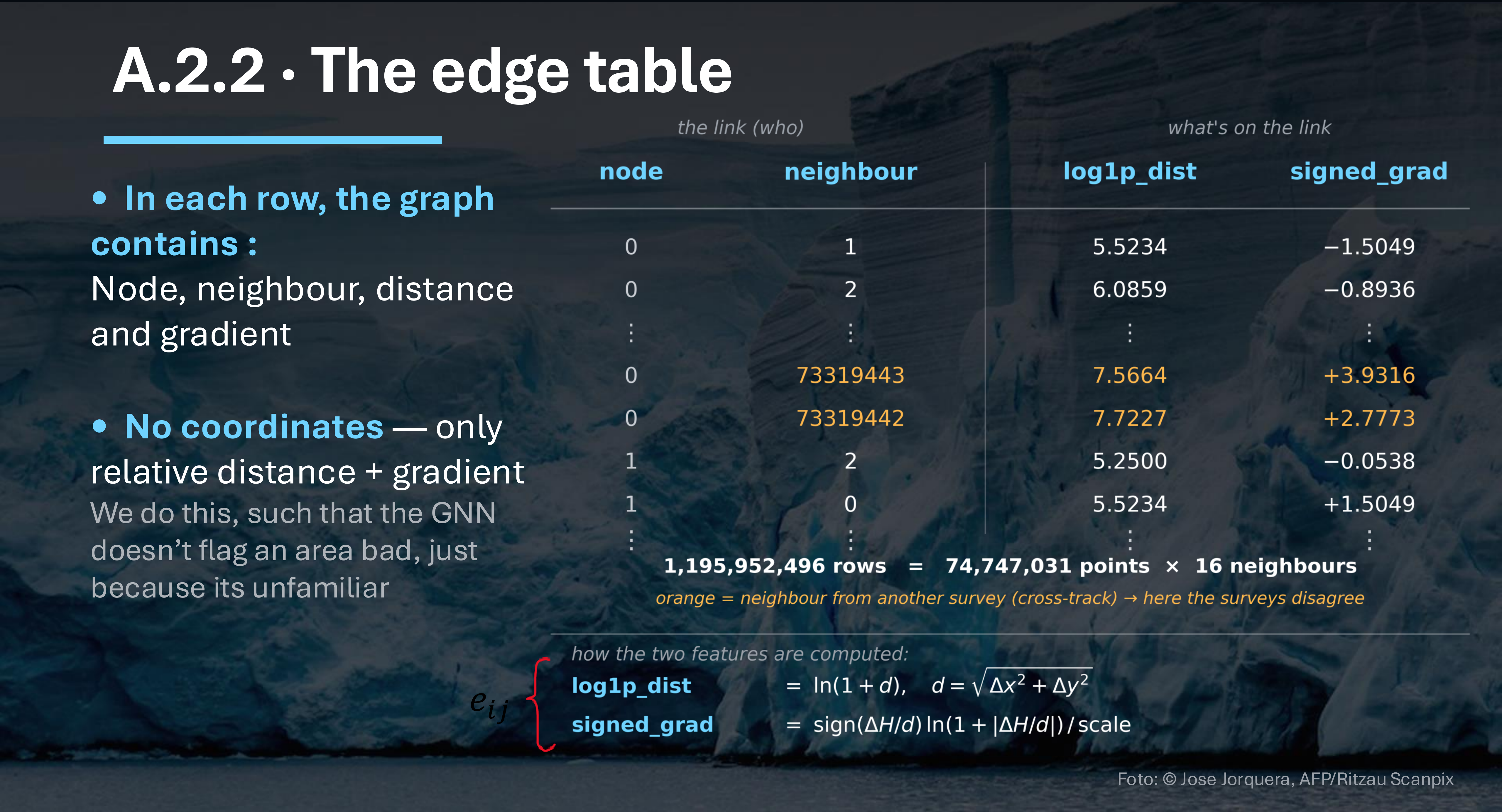
A.2.1 · The k-NN-16 map

How all 74.7M points are connected

- **Graph** — every point linked to its 16 nearest neighbours, by Euclidian distance $\left(\sqrt{x^2 + y^2}\right)$
- **≈ 1.2 billion links** (74.7M × 16)

every point → its 16 nearest neighbours

Foto: © Jose Jorquera, AFP/Ritzau Scanpix



A.2.2 · The edge table

- **In each row, the graph contains :** Node, neighbour, distance and gradient
- **No coordinates** — only relative distance + gradient
  We do this, such that the GNN doesn't flag an area bad, just because its unfamiliar

the link (who)		*what's on the link*	
node	**neighbour**	**log1p_dist**	**signed_grad**
0	1	5.5234	−1.5049
0	2	6.0859	−0.8936
⋮	⋮	⋮	⋮
0	73319443	7.5664	+3.9316
0	73319442	7.7227	+2.7773
1	2	5.2500	−0.0538
1	0	5.5234	+1.5049
⋮	⋮	⋮	⋮

1,195,952,496 rows = 74,747,031 points × 16 neighbours

orange = neighbour from another survey (cross-track) → here the surveys disagree

how the two features are computed:

$$e_{ij}\left\{\begin{array}{ll}\textbf{log1p\_dist} & = \ln(1+d), \quad d=\sqrt{\Delta x^2+\Delta y^2} \\ \textbf{signed\_grad} & = \operatorname{sign}(\Delta H/d)\ln(1+|\Delta H/d|)\,/\,\text{scale}\end{array}\right.$$

Foto: © Jose Jorquera, AFP/Ritzau Scanpix

A.3.1 · The training

- **Cross region 2-fold training :**

Fold A train nodes:
17,386 outliers
197,754 inliers
1,000,000 unlabeled

1 hour 34 minutes
656k parameters

Fold B train nodes:
17,608 outliers
319,586 inliers
1,000,000 unlabeled

1 hour 54 minutes
656k parameters

- **Hyperparameter optimization by Optuna:**

lr = 0.00102
weight decay = 3.18e-05
pos_weight = 3.92
Unl_out_weight = 0.00209
hidden = 384
latent = 192
dropout = 0.111
fanout = 16,8,8
batch size = 4096

Foto: © Jose Jorquera, AFP/Ritzau Scanpix

Latent-Space Outlier Carving

- **Pseudo-labels → Guided transformation→ Applied to 74 mio points →HDBSCAN in latent-space.**

Step 1:

The guided transformation pulls the two groups apart

Step 2:

From a few seeds, the scan grows to flag the whole outlier group

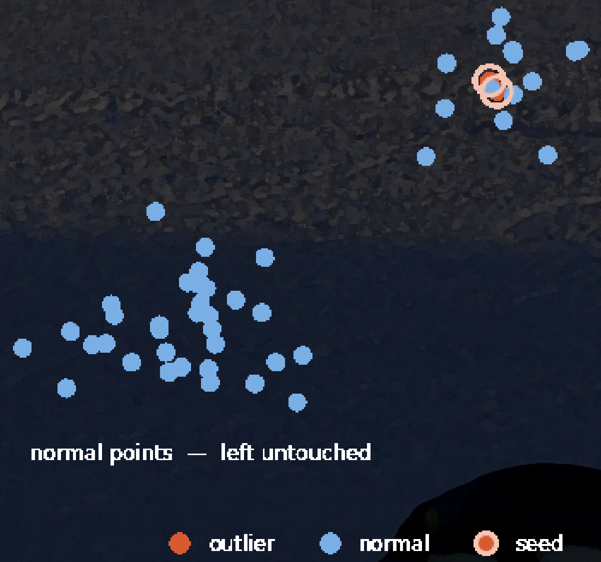
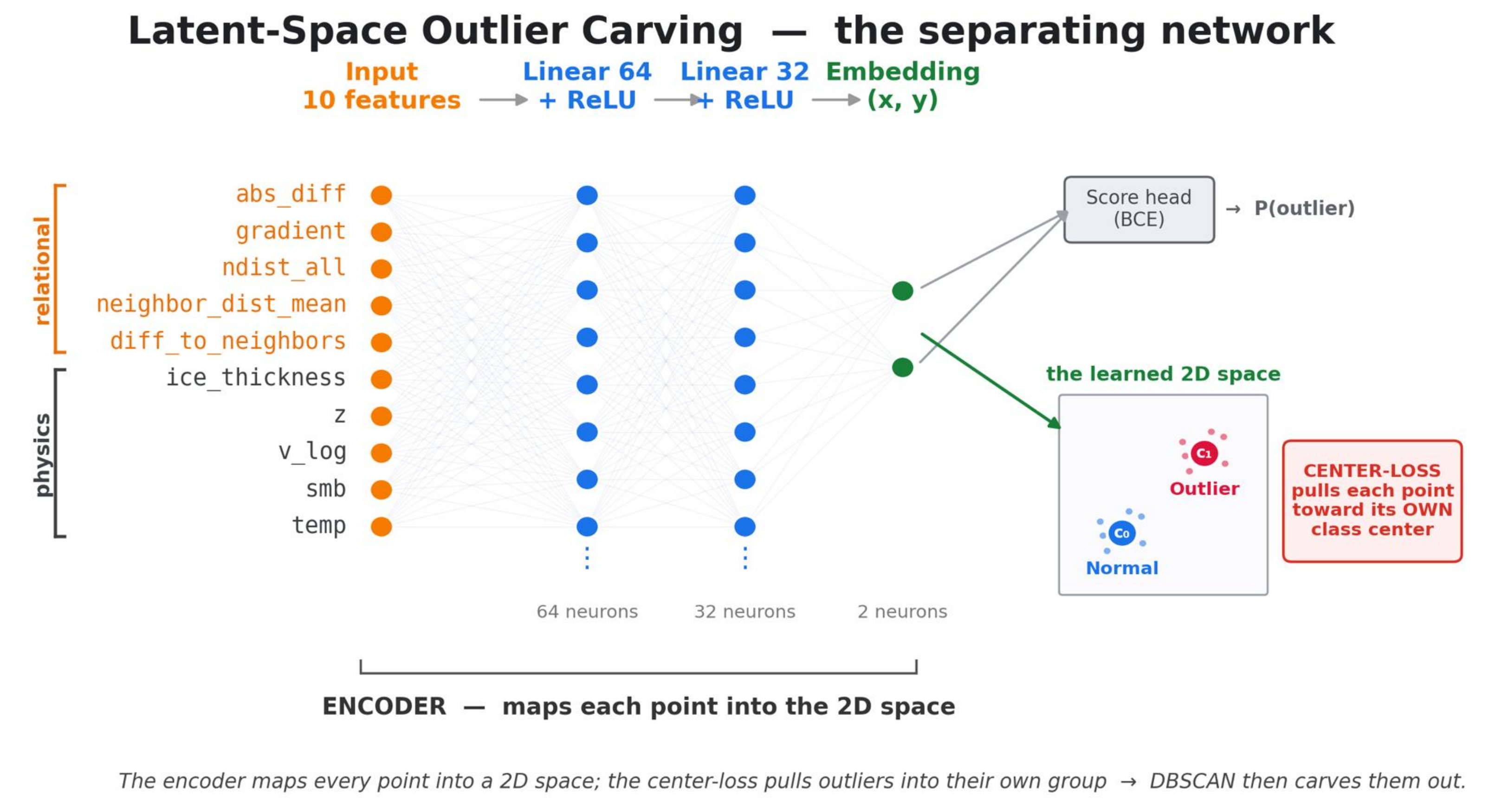


Foto: © Pixabay, Pexels.com

Latent-Space Outlier Carving

Guided Transformation



The encoder maps every point into a 2D space; the center-loss pulls outliers into their own group → DBSCAN then carves them out.

$$\mathcal{L} = \mathcal{L}_{BCE} - 1.0\mathcal{L}_{center} + 0.05\mathcal{L}_{repulsion}$$

Center-loss: labels pull each point to its class centre

L_{center}: pull point to its own centre

$L_{repulsion}$: push the two centres apart

L_{bce}: keep the boundary sharp

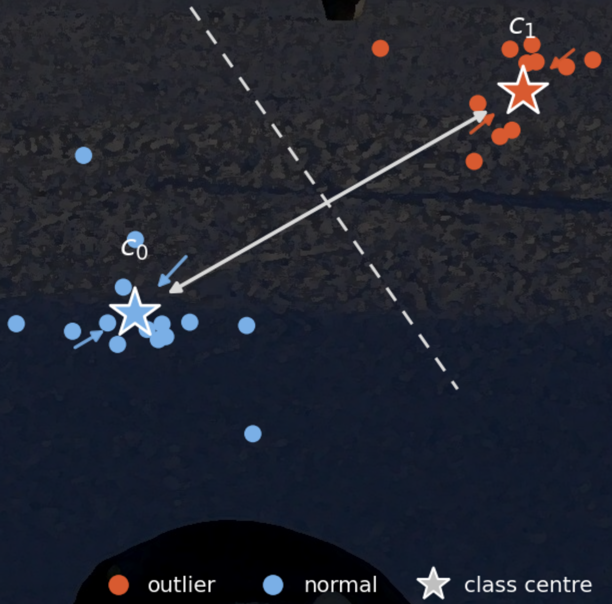


Foto: © Pixabay, Pexels.com

Latent-Space Outlier Carving

Result: Separation of the Training set

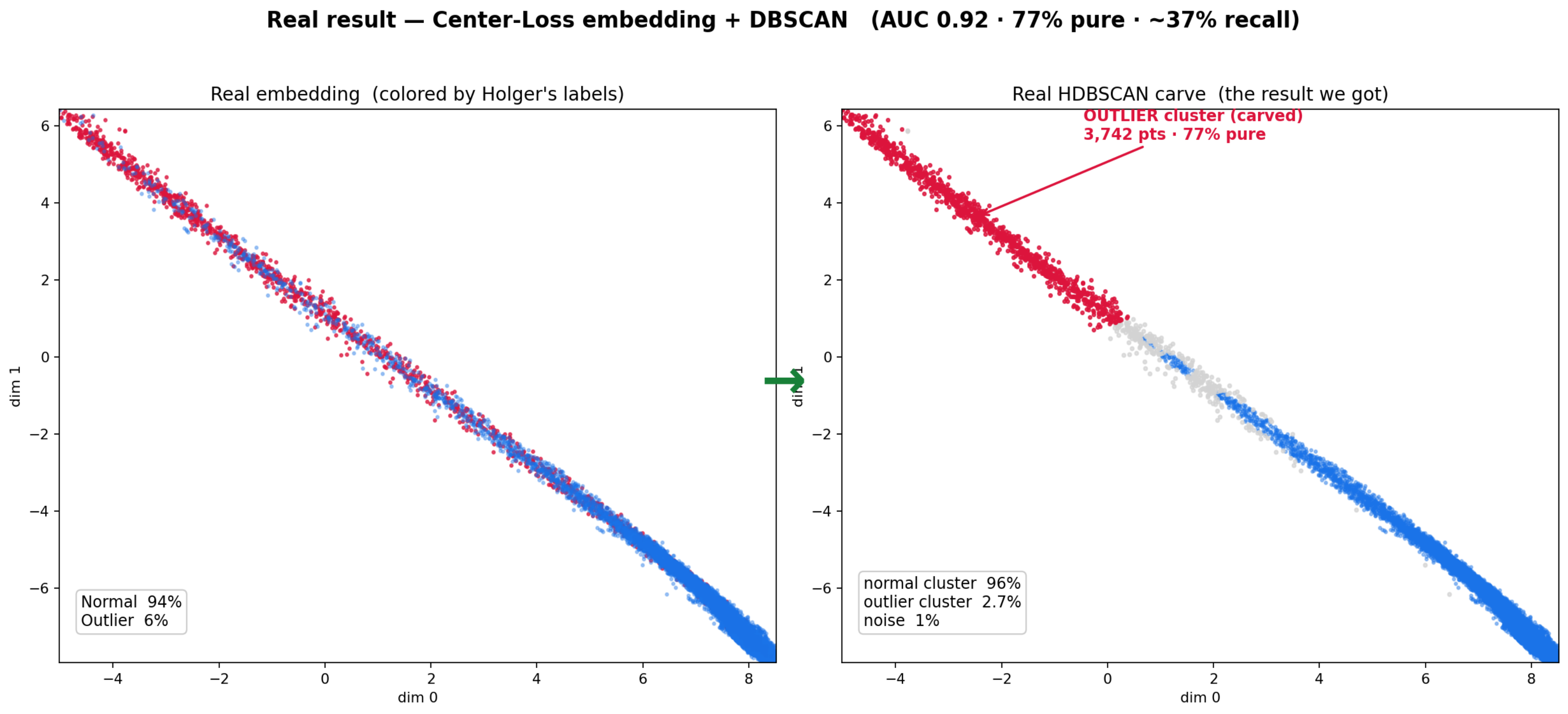


Foto: © Pixabay, Pexels.com

Latent-Space Outlier Carving

Applied to 74 Mio points and HDBSCAN

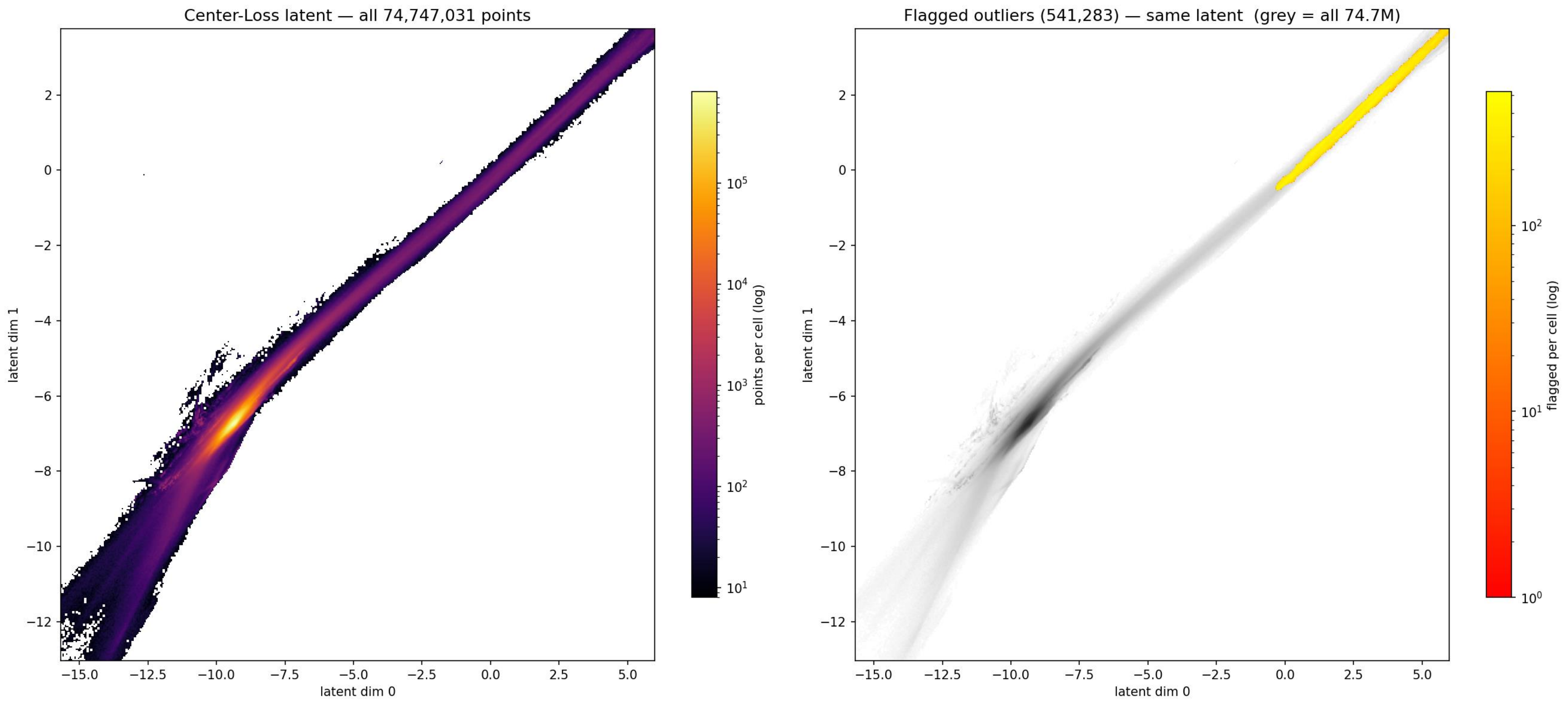
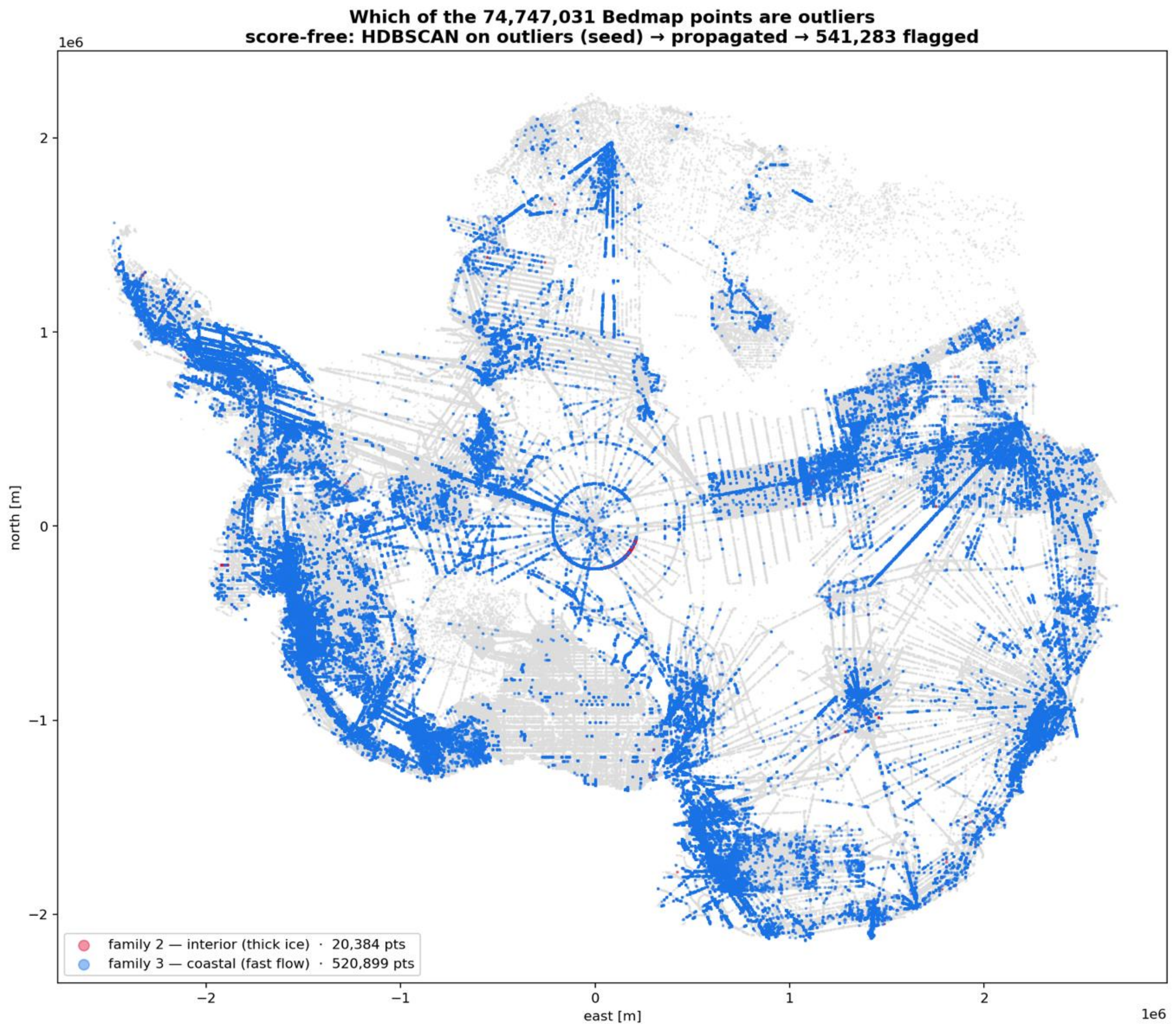


Foto: © Pixabay, Pexels.com

Latent-Space Outlier Carving

Final outlier map



- HDBSCAN: Creates families where dense clusters are!

Purity filter — which families are real outliers? (labels only, no score)

Family	Known outliers	Known normals in reach	Purity	Decision
2	624	5	**0.99**	✓ KEEP
3	10,374	3,385	**0.75**	✓ KEEP
0	240	348	**0.41**	✗ drop
1	284	618	**0.32**	✗ drop
4	211	871	**0.20**	✗ drop
5	266	1,467	**0.15**	✗ drop
6	12,671	453,686	**0.03**	✗ drop

Purity = outliers / (outliers + normals in reach). Keep a family only if purity ≥ 0.5.
Family 6 hides in the normal bulk (453,686 normals → purity 0.03) → dropped. Only families 2 & 3 survive → 541,283 clean outliers.

Foto: © Pixabay, Pexels.com

Graph – Autoencoder (AE)

- **Final outlier statistics:**

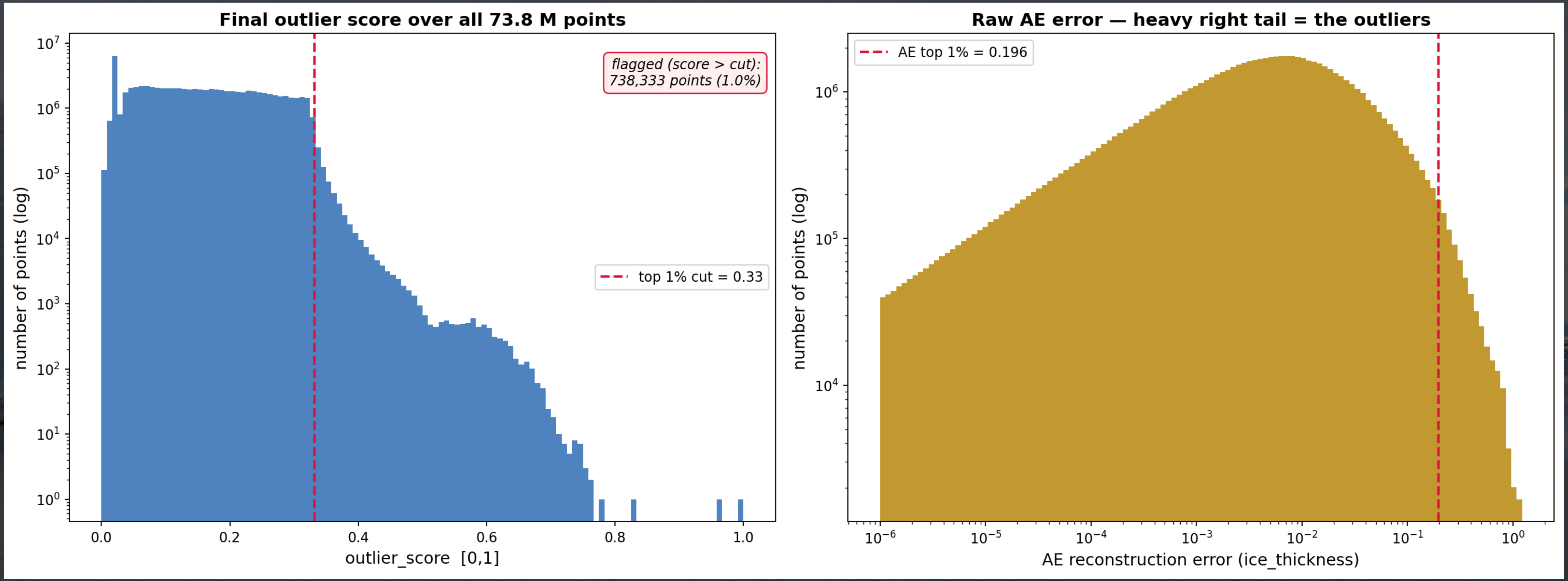
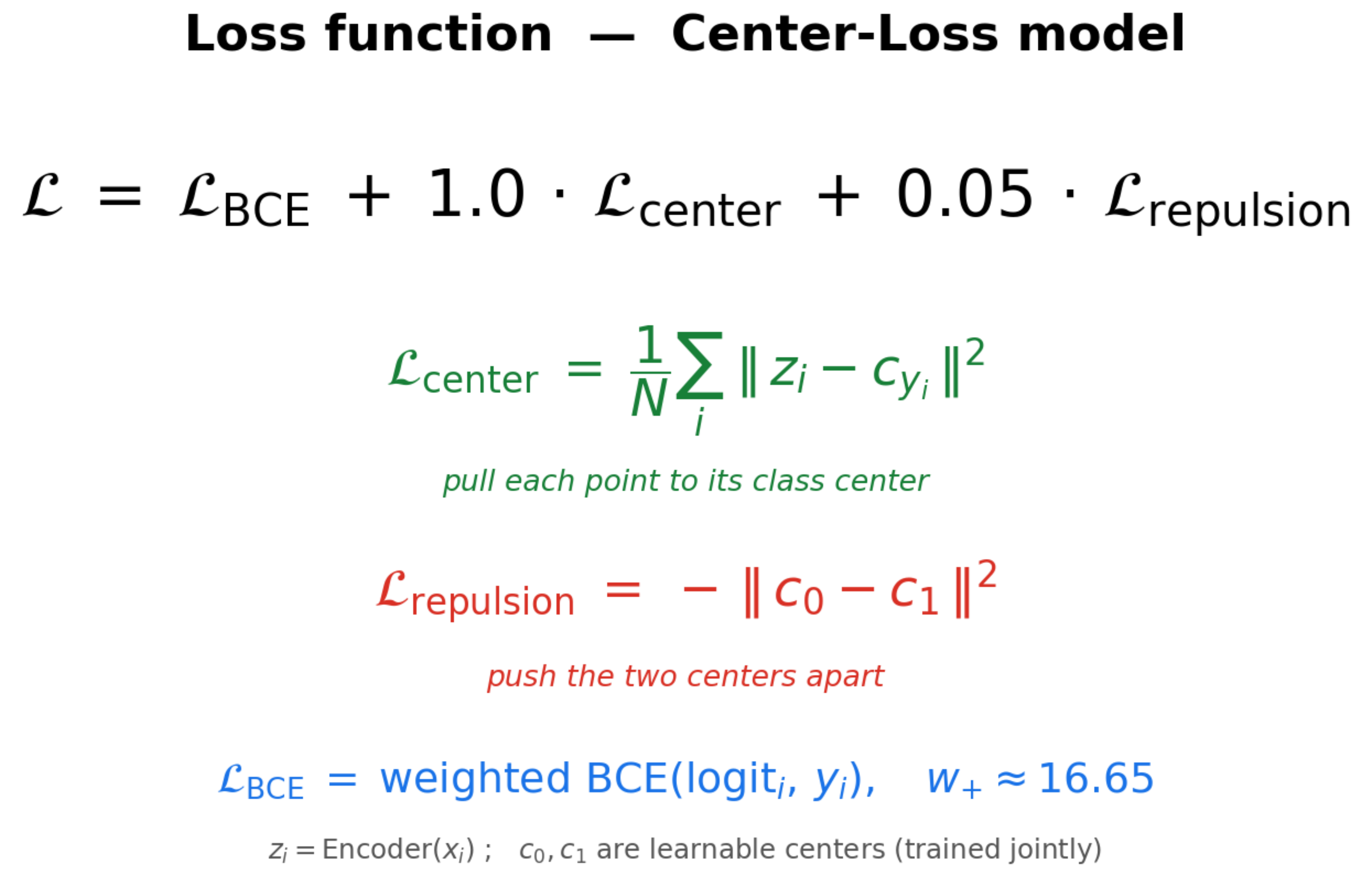


Foto: © ArcticDesire.com Polarreisen, Pexels.com

Latent-Space Outlier Carving – Loss Function



Loss function — Center-Loss model

$$\mathcal{L} = \mathcal{L}_{\text{BCE}} + 1.0 \cdot \mathcal{L}_{\text{center}} + 0.05 \cdot \mathcal{L}_{\text{repulsion}}$$

$$\mathcal{L}_{\text{center}} = \frac{1}{N}\sum_i \| z_i - c_{y_i} \|^2$$

pull each point to its class center

$$\mathcal{L}_{\text{repulsion}} = - \| c_0 - c_1 \|^2$$

push the two centers apart

$$\mathcal{L}_{\text{BCE}} = \text{weighted BCE}(\text{logit}_i,\, y_i), \quad w_+ \approx 16.65$$

$z_i = \text{Encoder}(x_i)$; c_0, c_1 are learnable centers (trained jointly)

Foto: © Pixabay, Pexels.com