PanoSwin: A Pano-Style Swin Transformer for Panorama Understanding

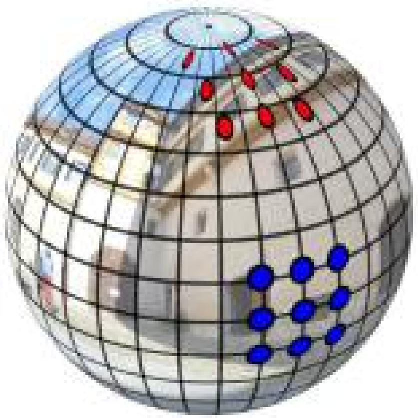
Zhixin Ling　Zhen Xing　Xiangdong Zhou　Manliang Cao　Guichun Zhou

School of Computer Science, Fudan University

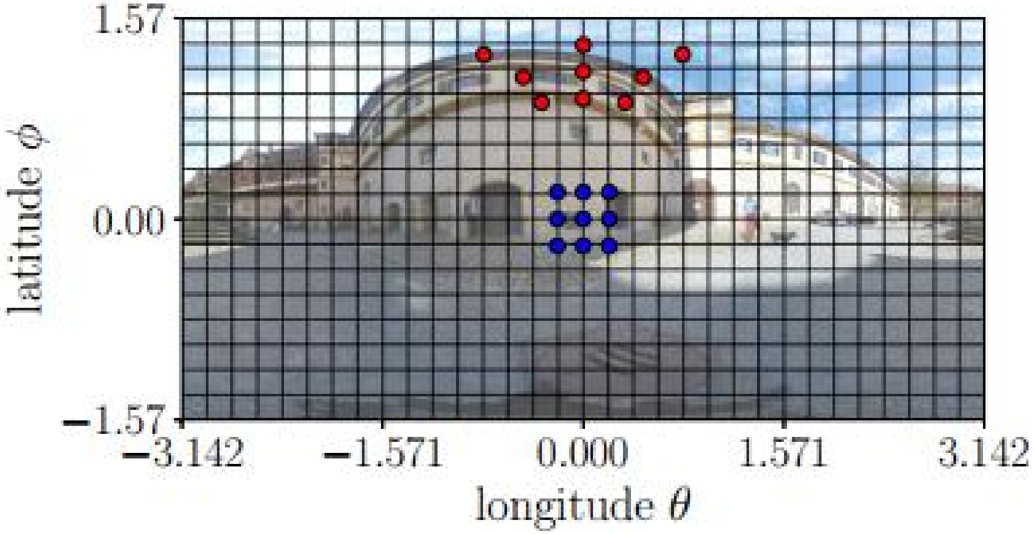
{20212010005, zxing20, xdzhou, 17110240029, 19110240014}@fudan.edu.cn

1. Background

2. Related Work: SphereNet



(a) Sphere

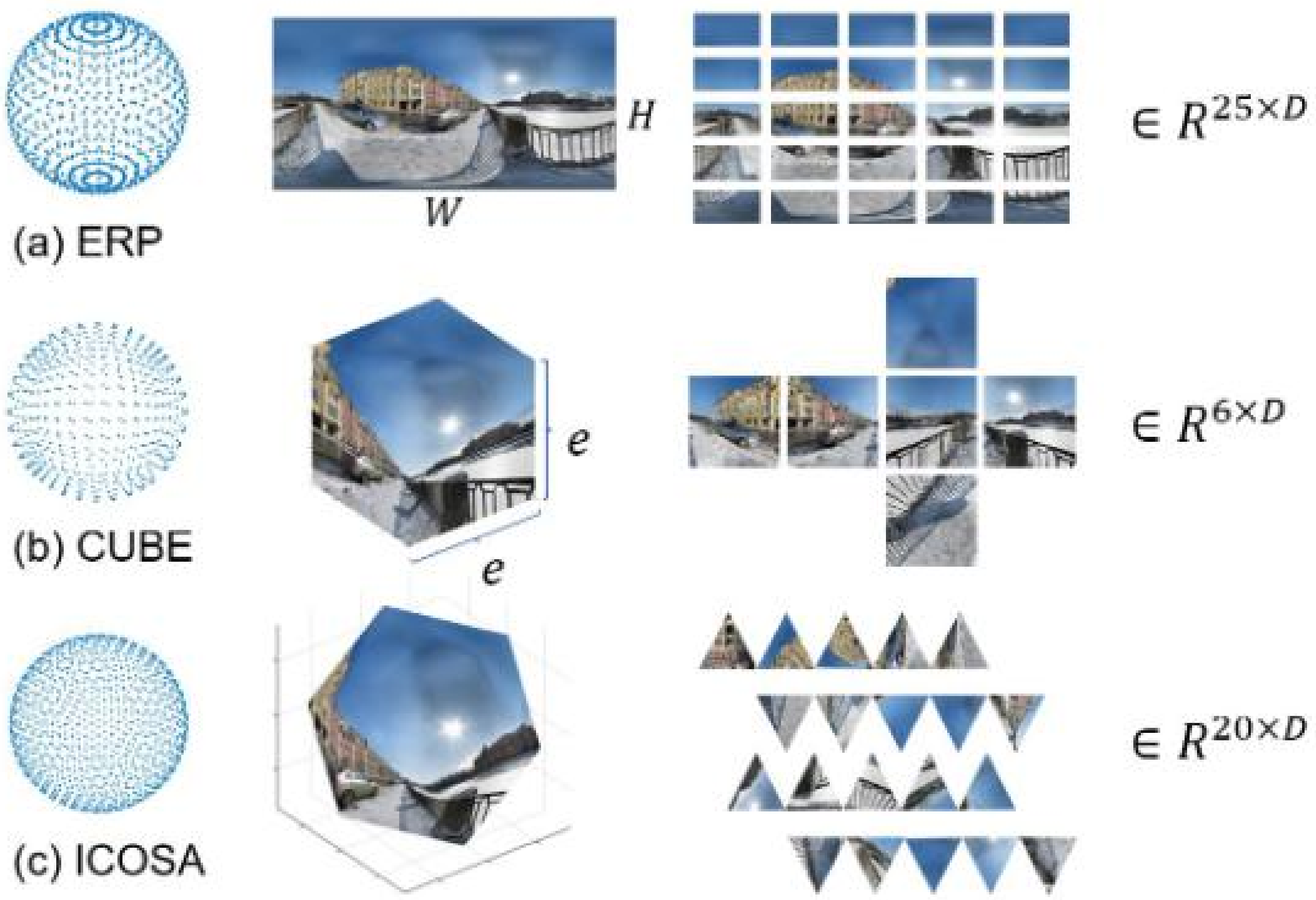


(b) Equirectangular

Strength: Project nearby pixels to a tangent plane, so regular CNNs can be adopted.

Weakness: Low parallelism, heavy computation overhead.

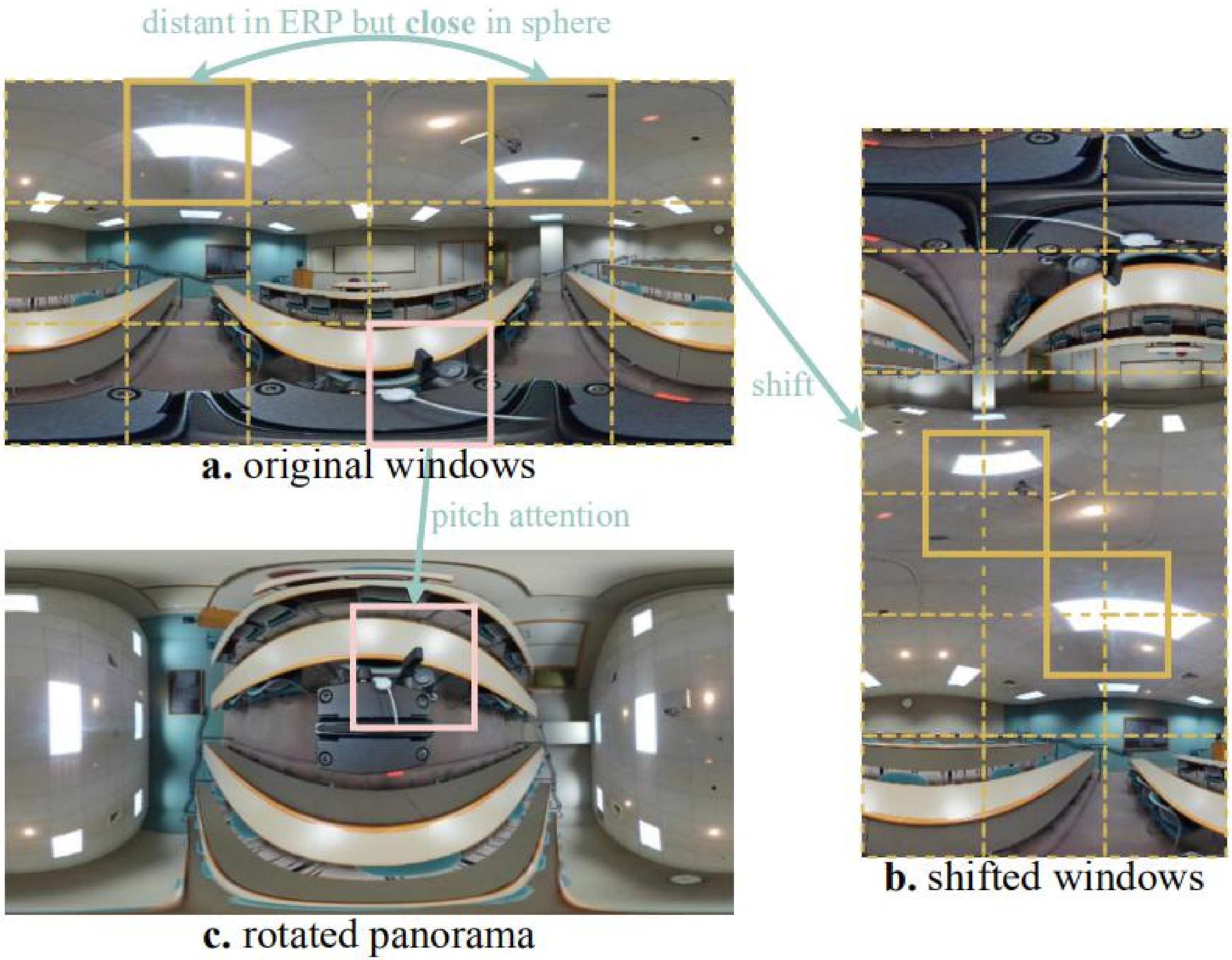
2. Related Work: Spherical Transformer



Strength: Resolve spatial distortion and discontinuity.

Weakness: Imperfect projection; unfeasible to planar images.

3. Our method: Overview of PanoSwin



1. Side *boundary discontinuity* can be overcome by removing the attention masks.
2. **a.** => **b.** : our pano-style shift windowing scheme overcomes polar *boundary discontinuity.*
3. **a.** => **c.** : Pitch Attention lets a distorted window to "see" its original appearance to resolve *spacial distortion*.

3. Our method: A pano-style shift windowing scheme

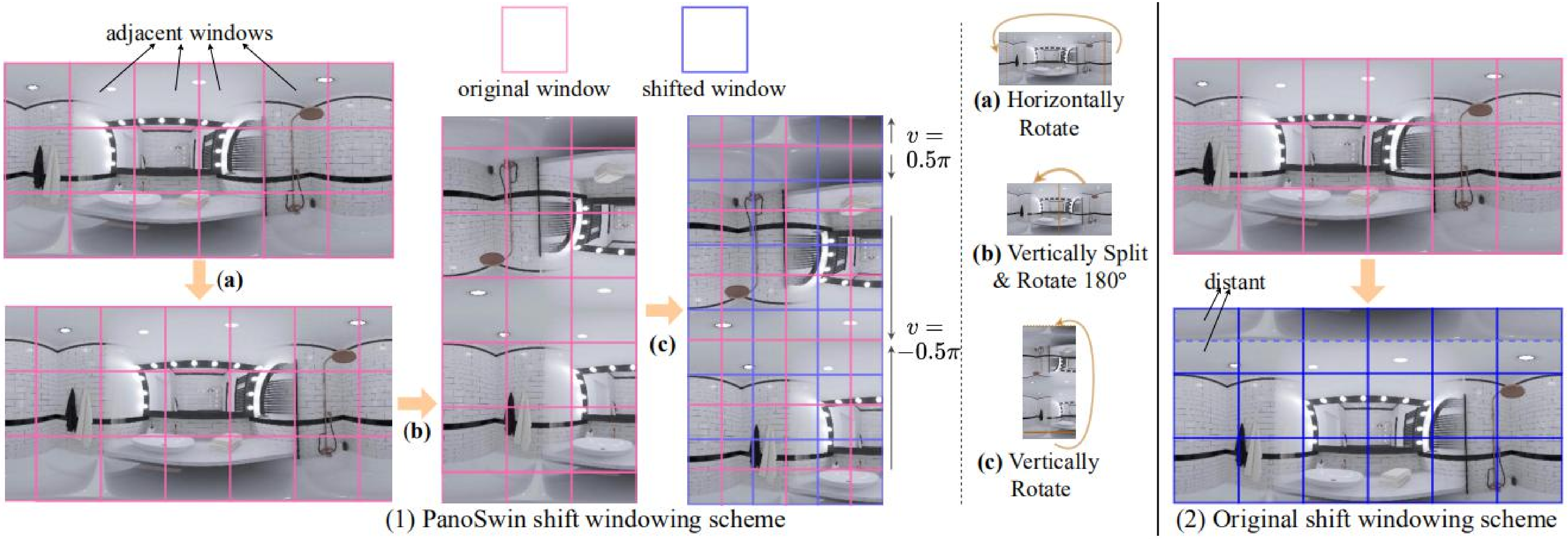
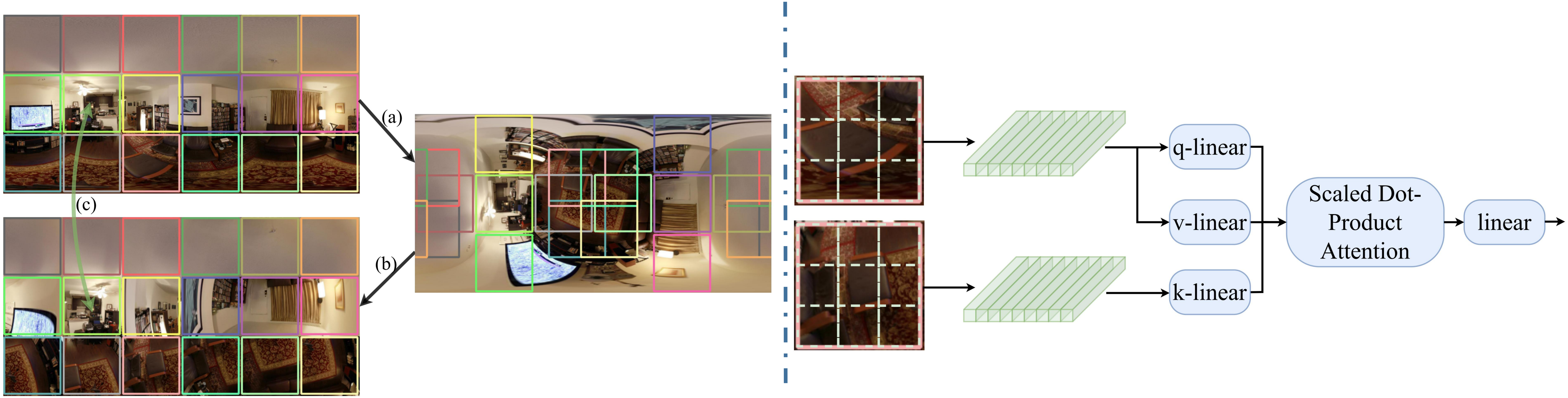


Figure 2. Pano-style/original shift windowing scheme comparison. The arrowed line in orange shows each conversion step.

Pano-style Shift Windowing scheme (PSW) consists of three steps:

1. Horizontally shift the image to enable the left/right side continuity.
2. Split the image in half and rotate the right half by 180° counterclockwise to enable the north pole continuity.
3. Vertically shift the image to enable the south pole continuity.

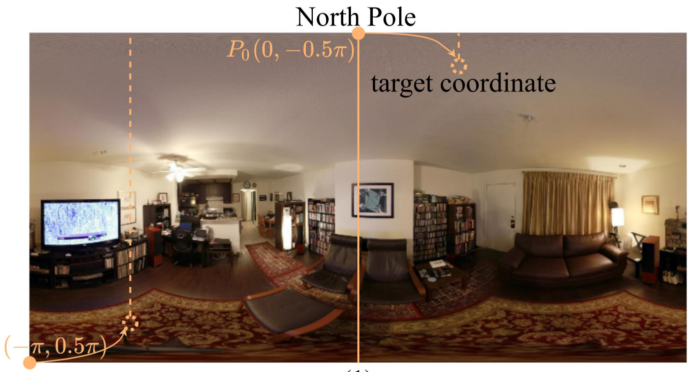
3. Our method: Pitch Attention



Pitch Attention module (PA) consists of three steps:

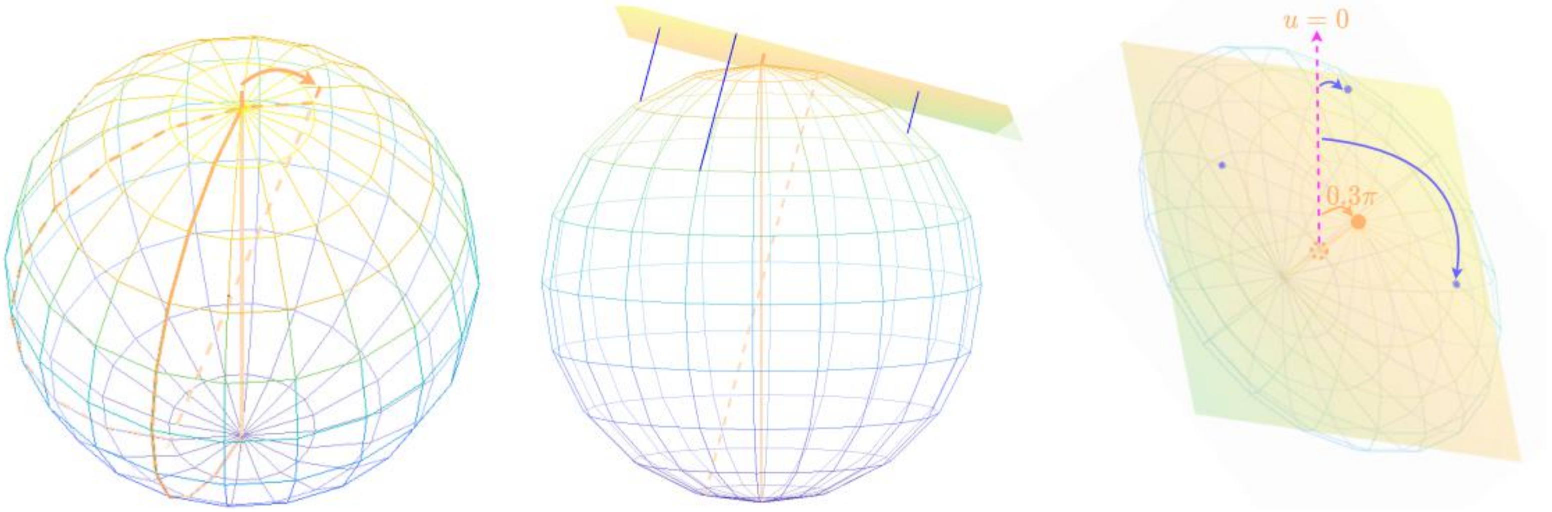
1. Rotate the pitch of the panorama by 90°.
2. Sample a new window in the rotated panorama for each original window..
3. Perform window attention between original and new windows.

3. Our method: Panoramic Rotation



(1)

(2)



(3)

Sph(P) gives the Cartesian coordinate for a point P.

we can explain the function R in a formula:

$$\begin{aligned} v' &= 2\text{asin}(\frac{1}{2}||\text{Sph}(P) - \text{Sph}(P_1)||_2) - 0.5\pi, \\ P_a \hat{\otimes} P_b : & \quad \text{Sph}(P_a) \otimes \text{Sph}(P_b), \\ u' &= \text{Angle}(P \hat{\otimes} P_1, P_0 \hat{\otimes} P_1, (P_0 \hat{\otimes} P_1) \otimes P_1), \end{aligned} \tag{2}$$

3. Our method: Two-stage Learning Paradigm

Algorithm 1: two-stage learning paradigm.

Input: a downstream task loss $\mathcal{L}_{DS}$; a randomly initialied PanoSwin model $\mathcal{P}$.
Output: A trained PanoSwin model.

$\mathcal{A}^{plan} \leftarrow$ a set of planar augmentation methods, *e.g.*, random resizing, cropping and rotation;
$\mathcal{A}^{pano} \leftarrow$ a set of pano-compatible augmentation methods, *e.g.*, random panoramic rotation, flipping, color jittering;
Define $train(model, loss, augs)$ as a function that trains $model$ by optimizing $loss$ and enables augmentation approaches specified by $augs$;
$\mathcal{T} \leftarrow train(model = \mathcal{P}_s, loss = \mathcal{L}_{DS}, augs = \mathcal{A}^{plan} \cup \mathcal{A}^{pano})$;
$\mathcal{S} \leftarrow \mathcal{T}$; $fix(\mathcal{T})$; $fix(\alpha_{i,j}$ of $\mathcal{S})$; $\mathcal{S} \leftarrow train(model = \mathcal{S}_p, loss = \mathcal{L}_{DS} + \mathcal{L}_{KP}, augs = \mathcal{A}^{pano})$;
return $\mathcal{S}$

$$\mathcal{L}_{KP} = \frac{1}{\sum_i^N w_i} \sum_i^N w_i ||A(\mathcal{S}(x))^{(i)} - \mathcal{T}_s(x)^{(i)}||_2^2, \text{ where } w_i = \cos^2(v_i)\cos^2(\tfrac{1}{2}u_i)$$

Note that PanoSwin is divised to be compatible with planar images, so common knowledge can be easily transferred from planar images to panoramas via a two-stage learning paradigm and a KP loss.

4. Results: Qualitative Comparison

SPH-Cifar10 classification

No.	Backbone	acc↑	para.
C1	SpherePHD [16]	59.20	57k
C2	SphericalTransformer [2]	58.21	60k
C3	SGCN [34]	60.72	60k
C4	S2CNN [4]	10.00	58k
C5	SwinT13 [19]	60.46	67k
C6	PanoSwinT12	**62.24**	66k
C7	SwinT [19]	72.64	28M
C8	PanoSwinT92	74.50	28M
C9	PanoSwinT	74.84	30M
C10	PanoSwinT$^+$	**75.01**	30M

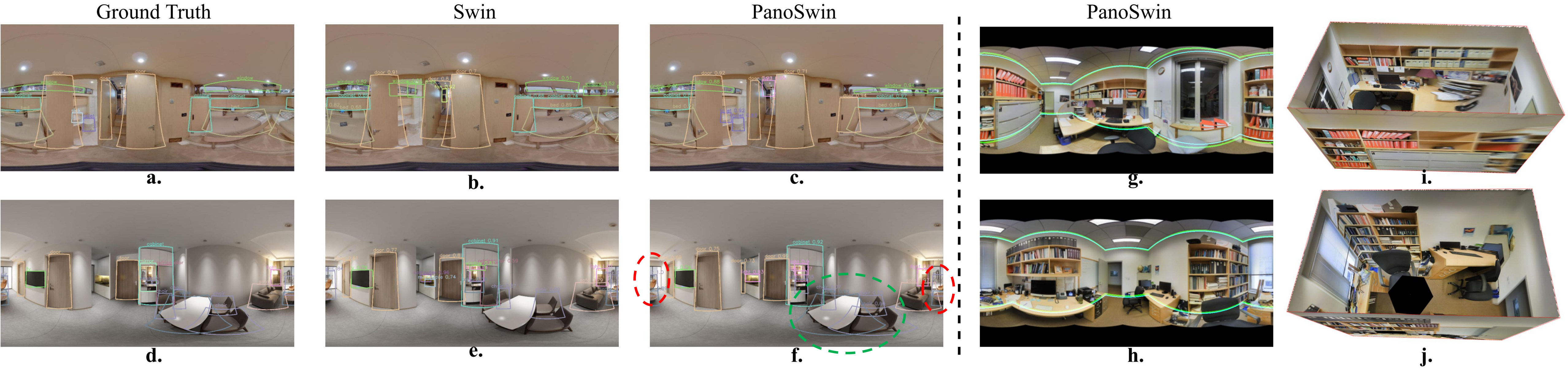
360Indoor Object detection

No.	Backbone	mAP@0.5↑	para.
I1	R50 [11] + COCO	33.1	72M
I2	SwinT [19] + COCO	33.8	45M
I3	PanoSwinT92 + COCO	35.6	45M
I4	R50 [11]	20.6	72M
I5	R50 [11] + SC [5]	21.1	72M
I6	SwinT [19]	24.0	45M
I7	PanoSwinT92	28.0	45M
I8	PanoSwinT	28.6	47M
I9	PanoSwinT$^+$	**29.4**	47M

Inference Time

	PST	PST$_s$	SwinT	KTN [24]	PST8	SN
para.	30M	30M	28M	294M	191k	196k
CPU↓	1.207	1.018	0.982	5.136	0.186	0.682
GPU↓	0.042	0.015	0.010	3.842	0.021	0.025

4. Results: Qualitative Comparison



Please notice the *spatial distortion* and *boundary discontinuity*

Thanks!