DarSwin-Unet: Distortion Aware Encoder Decoder Architecture

Leonardo Sacht

360 e2e: Analysis and Synthesis of Omnidirectional Video

November 13th, 2025

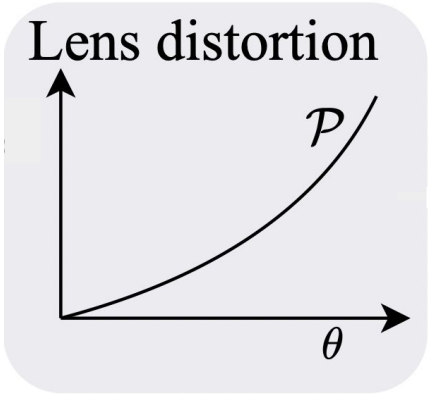
Complete reference

DarSwin-Unet: Distortion Aware Encoder Decoder Architecture

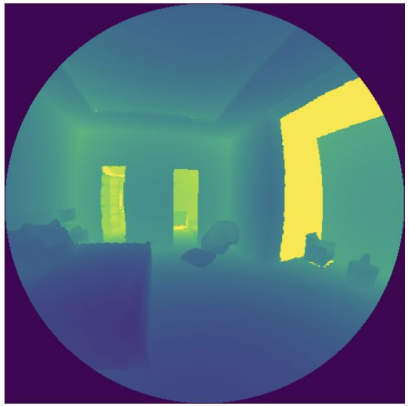
Authors: Akshaya Athwale, Ichrak Shili, Émile Bergeron, Ola Ahmad, Jean-François Lalonde (Université Laval)

Conference: 2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)

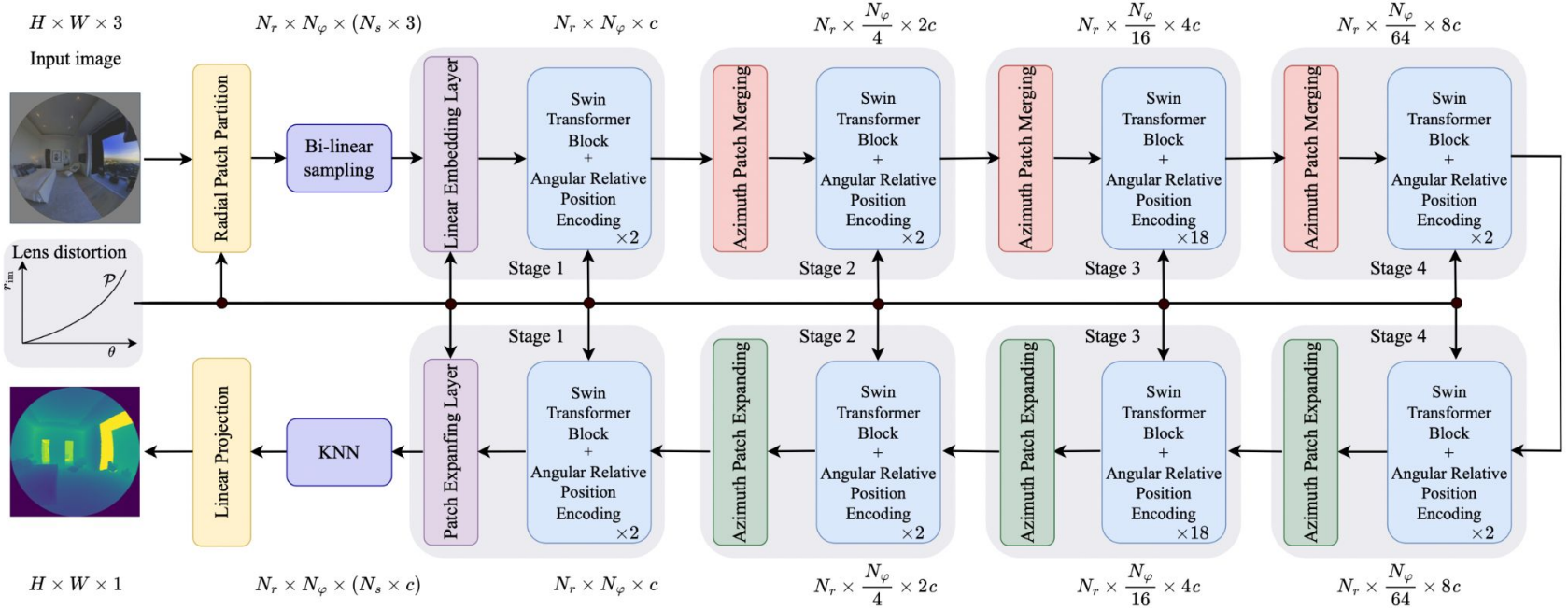
Goal: wide-angle image pixel-level tasks



...

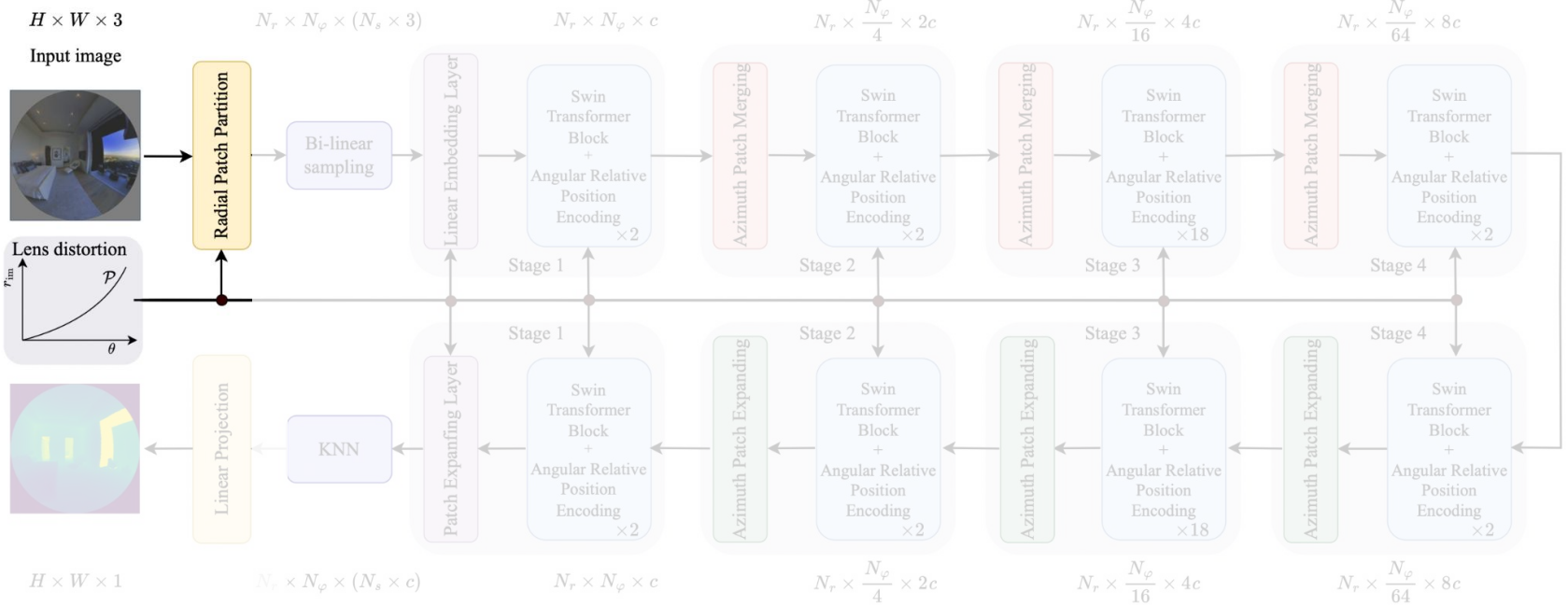


Contributions of DarSwin-Unet



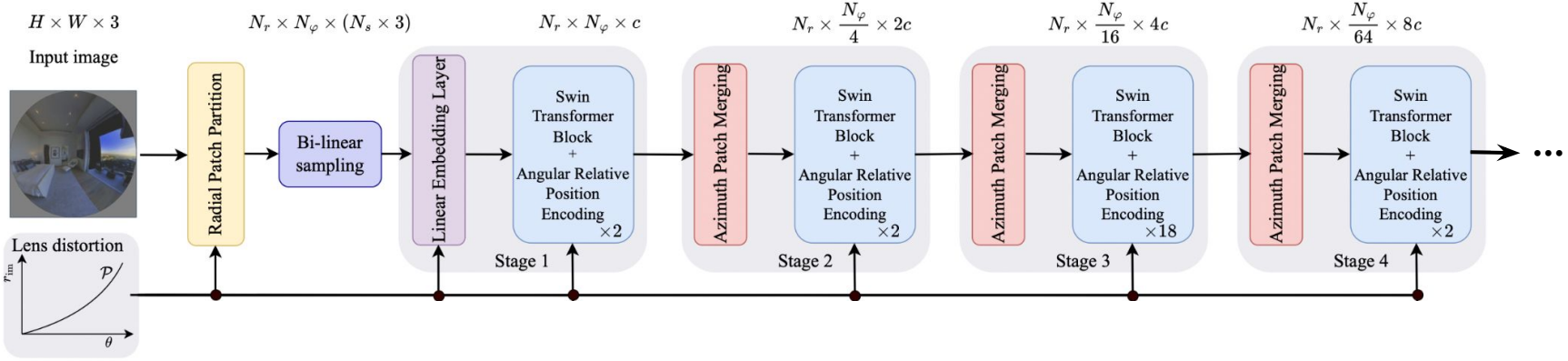
- Encoder-**decoder** transformer network

Contributions of DarSwin-Unet



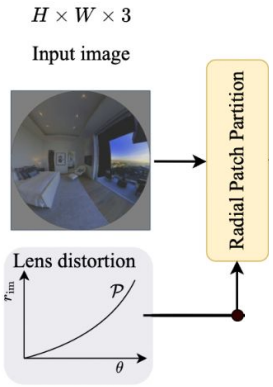
- Encoder-**decoder** transformer network
- New pixel sampling scheme

DarSwin (2023): Encoder-only



- Distortion-aware
- Non-pixel-level tasks (e.g., classification)

DarSwin (2023): distortion-aware partition



(Known) Lens distortion:

$$r_d = \mathcal{P}(\theta) = \frac{f \cos\theta}{\xi + \sin\theta}$$

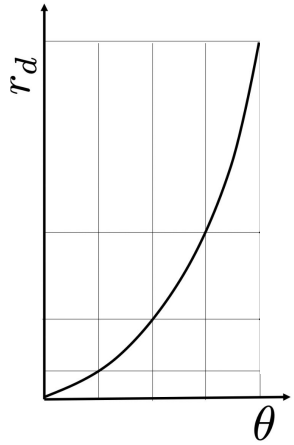
DarSwin (2023): distortion-aware partition

(Known) Lens distortion:

$$r_d = \mathcal{P}(\theta) = \frac{f \cos\theta}{\xi + \sin\theta}$$

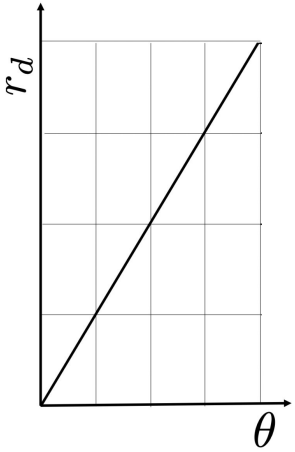
$\xi = 0$

("Low" distortion)



$\xi = 1$

("High" distortion)



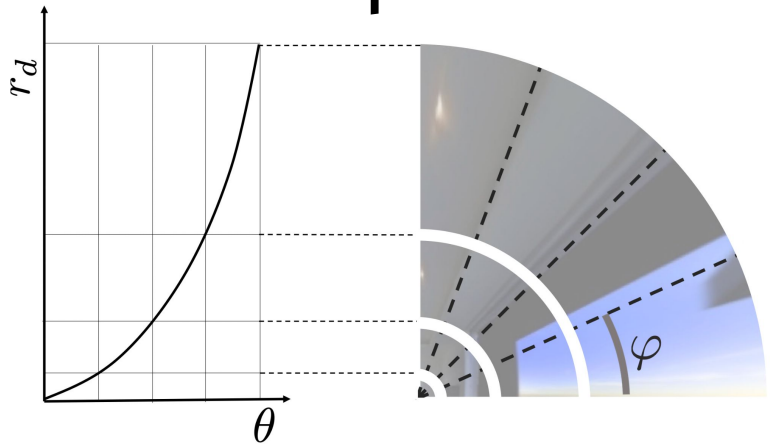
DarSwin (2023): distortion-aware partition

(Known) Lens distortion:

$$r_d = \mathcal{P}(\theta) = \frac{f \cos\theta}{\xi + \sin\theta}$$

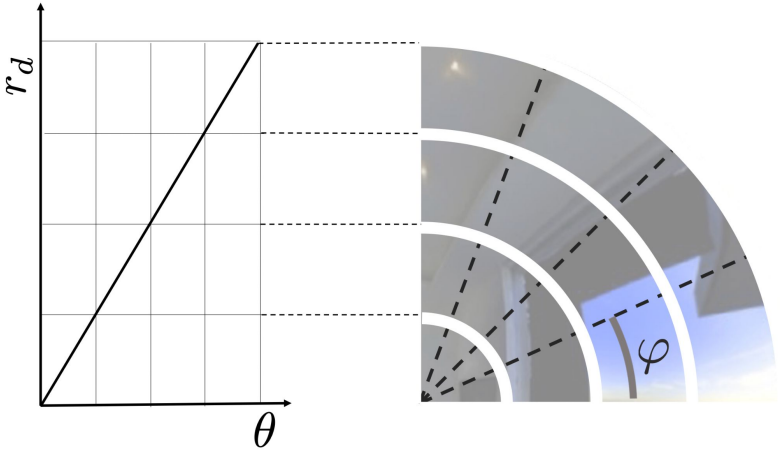
$\xi = 0$

("Low" distortion)

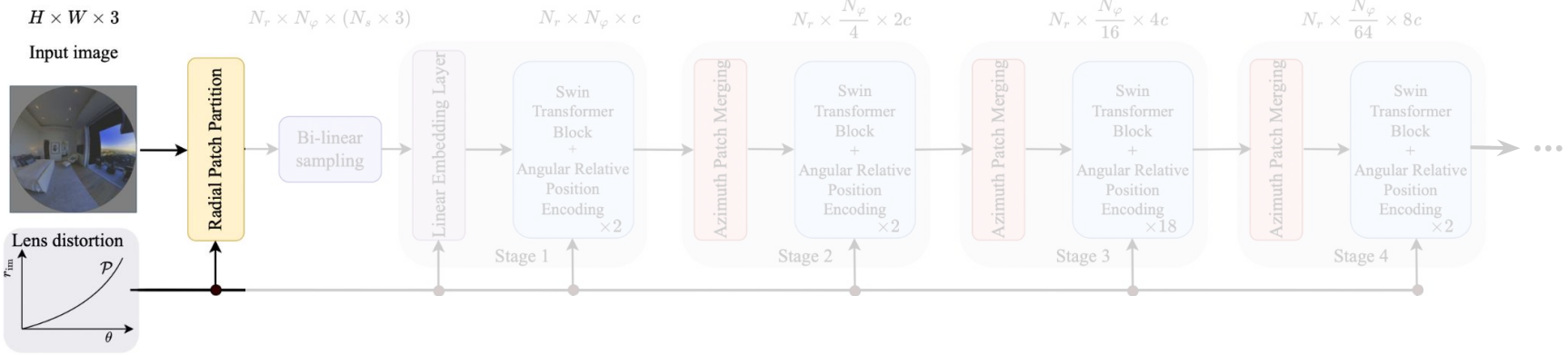


$\xi = 1$

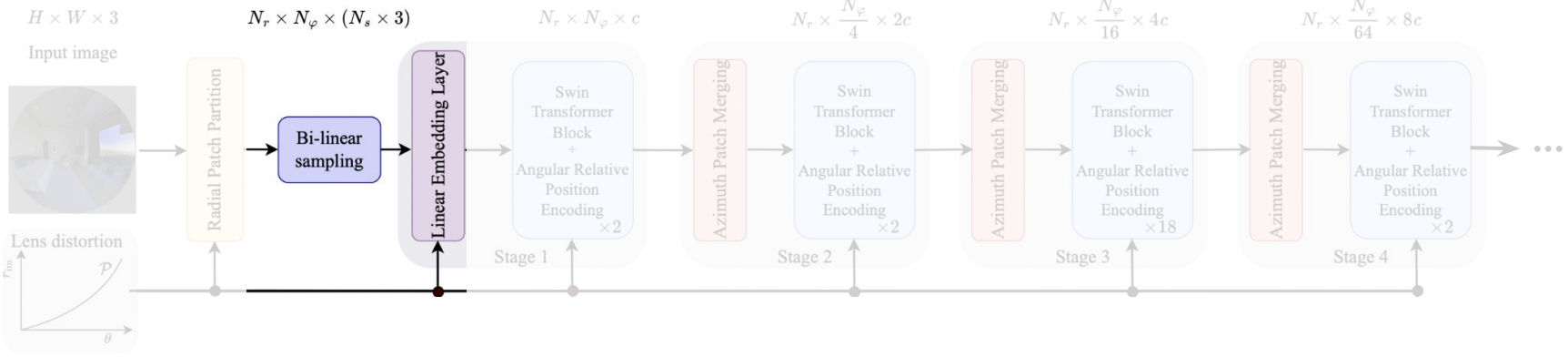
("High" distortion)

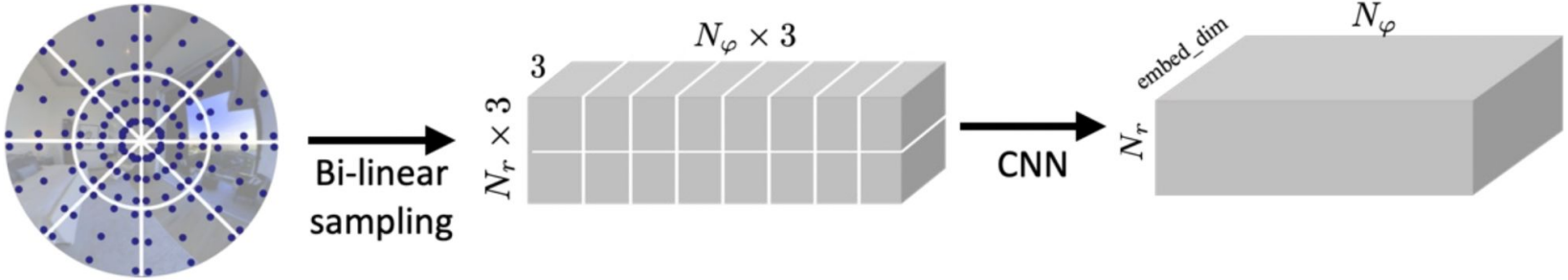


DarSwin (2023)

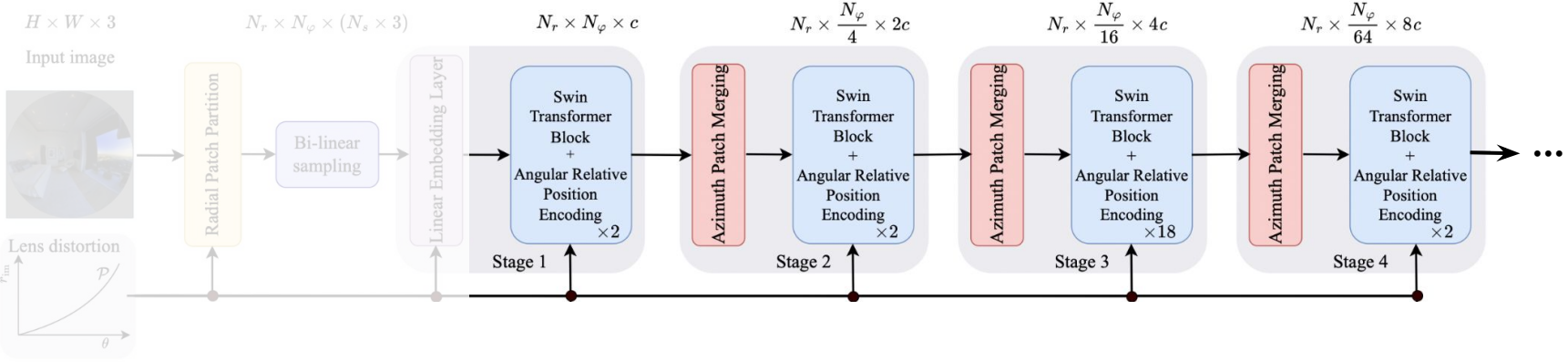


DarSwin (2023): sampling and CNN



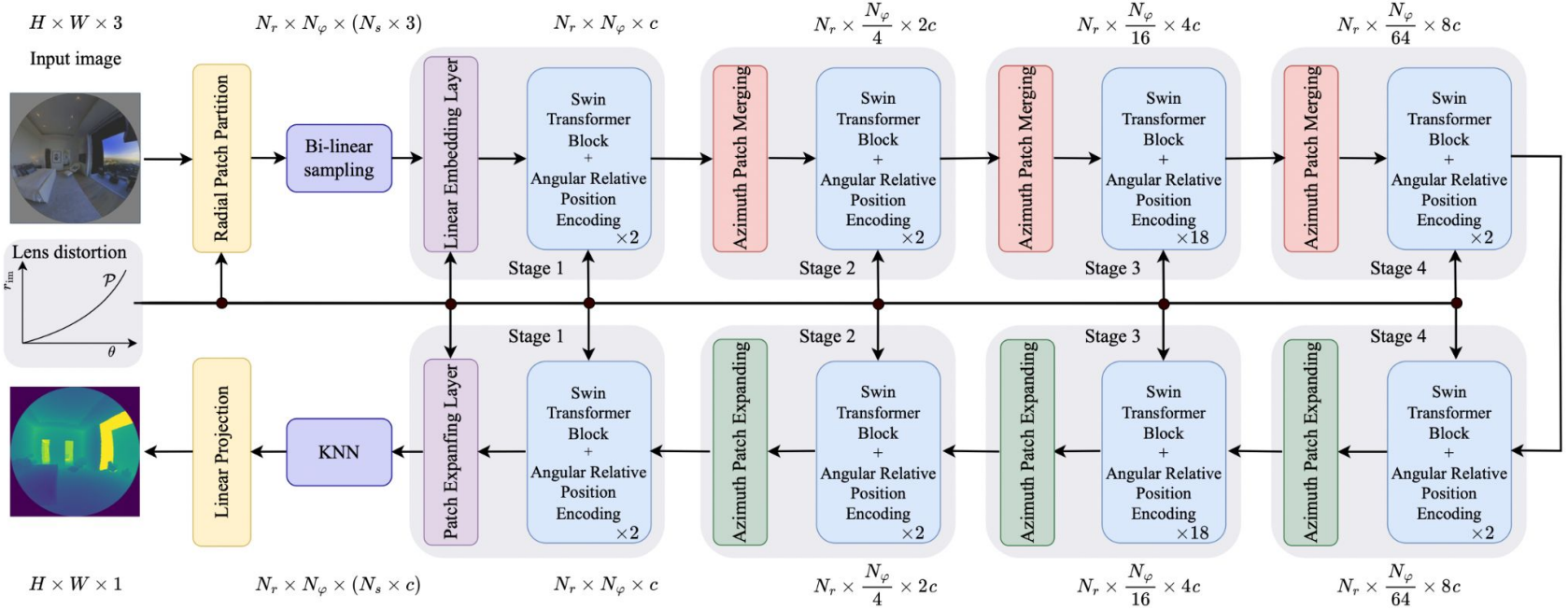


DarSwin (2023): transformers and merging



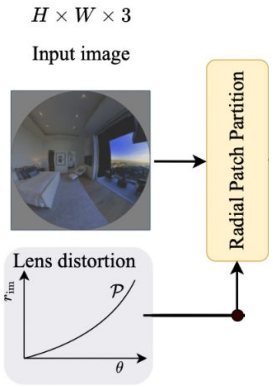
- A set of non-overlapping shifted windows is defined using the patches along the azimuth dimension.
- Shifts are obtained by displacing the windows along the azimuth.
- Downsampling to merge groups of four angular patches.

DarSwin-Unet (2025): encoder-decoder



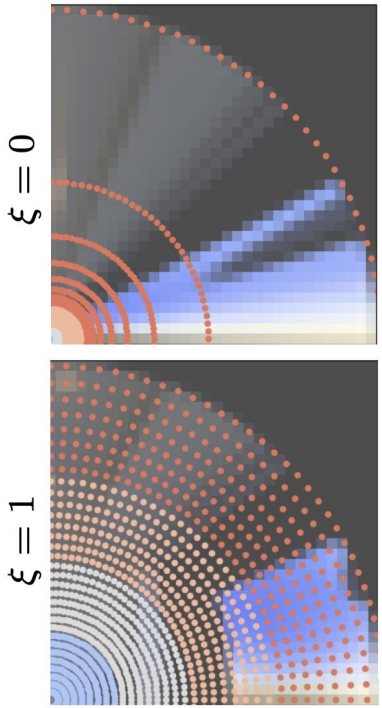
- Patch partition improvement
- Decoder to perform pixel-level tasks

DarSwin (2023): sparsity of samples

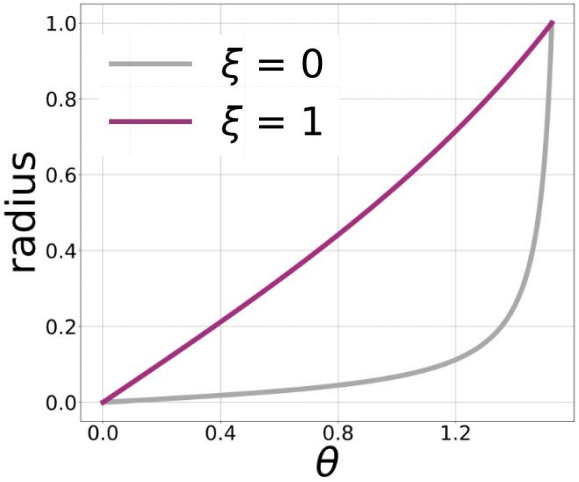


(Known) Lens distortion:

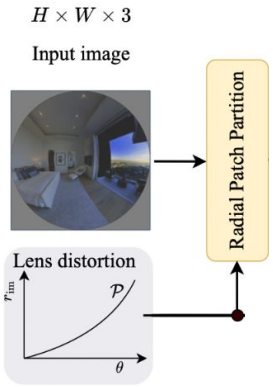
$$r_d = \mathcal{P}(\theta) = \frac{f \cos\theta}{\xi + \sin\theta}$$



(a) $r_d = \mathcal{P}(\theta)$

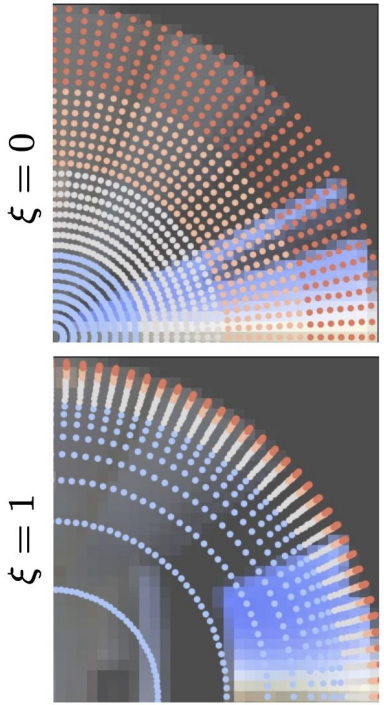


Sparsity of samples

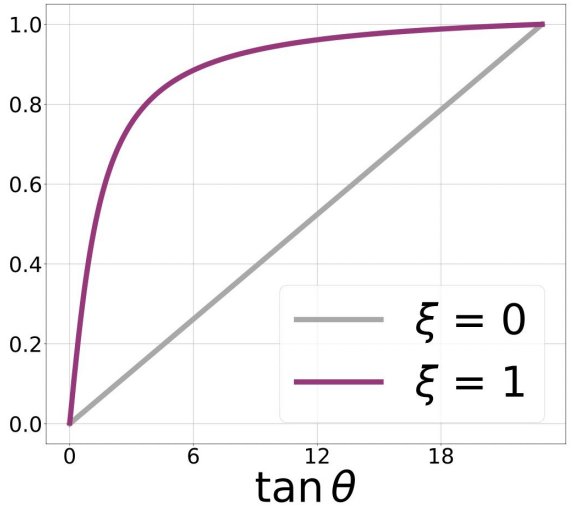


(Known) Lens distortion:

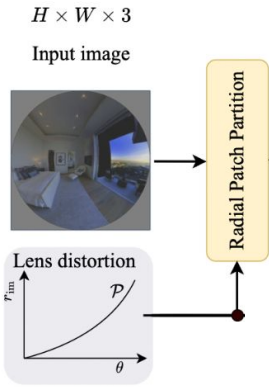
$$r_d = \mathcal{P}(\theta) = \frac{f \cos\theta}{\xi + \sin\theta}$$



(b) $r_d = \mathcal{P}(\tan\theta)$

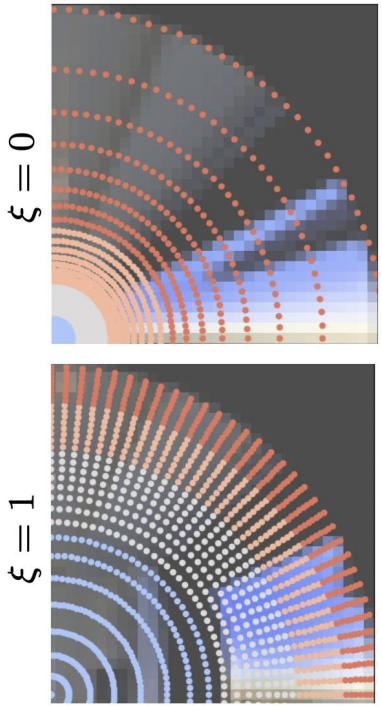


DarSwin-Unet (2025): better sampling

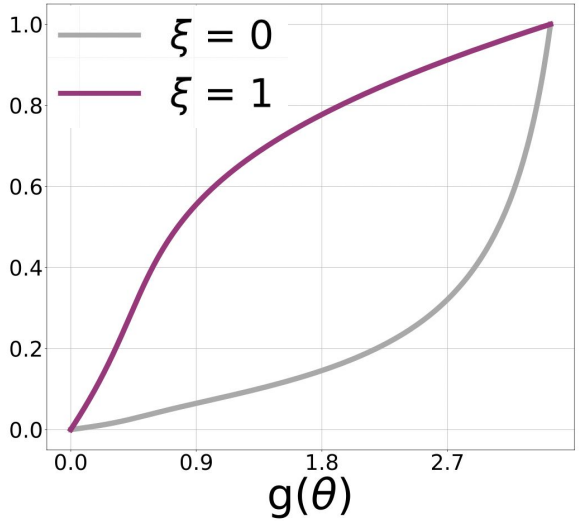


(Known) Lens distortion:

$$r_d = \mathcal{P}(\theta) = \frac{f \cos\theta}{\xi + \sin\theta}$$



(c) $r_d = \mathcal{P}(g(\theta))$



DarSwin-Unet (2025): better sampling

- Objective function:

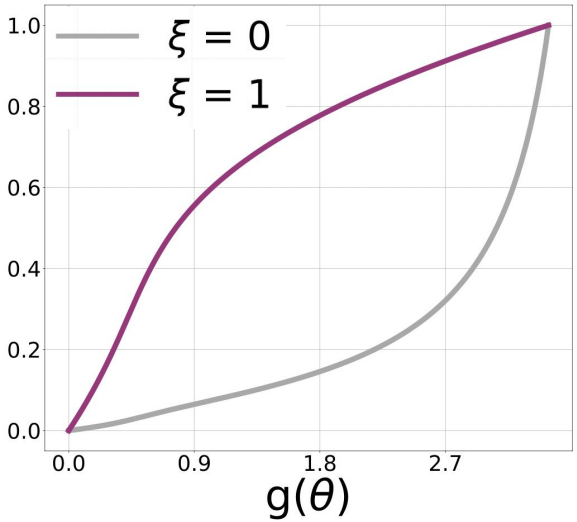
$$\max_{\theta}\left(\frac{d\mathcal{P}(g(\theta))|_{\xi=0}}{d(g(\theta))}\right)+\max_{\theta}\left(\frac{d\mathcal{P}(g(\theta))|_{\xi=1}}{d(g(\theta))}\right)$$

- Search space:

$$g(\theta)=\lambda p_n(\theta)+(1-\lambda)q_m(\theta)\,,$$

$$\text{with } p_n(\theta)=b\left(\frac{\theta}{a}\right)^n \text{ and } q_m(\theta)=1-\left(1-\frac{\theta}{a}\right)^m$$

- Exhaustive search on parameters



DarSwin-Unet (2025): better sampling

- Objective function:

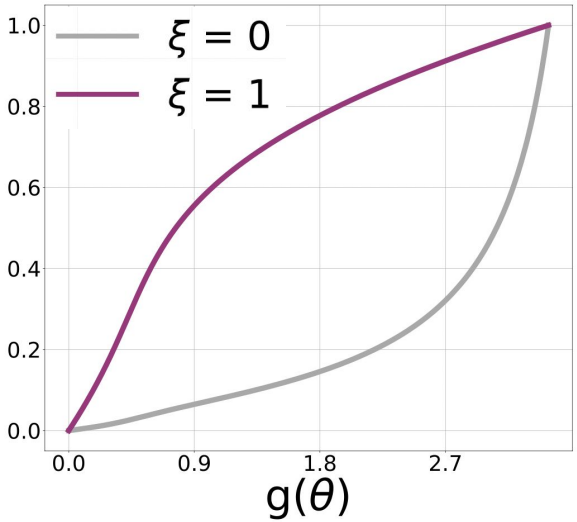
$$\max_{\theta}\left(\frac{d\mathcal{P}(g(\theta))|_{\xi=0}}{d(g(\theta))}\right)+\max_{\theta}\left(\frac{d\mathcal{P}(g(\theta))|_{\xi=1}}{d(g(\theta))}\right)$$

- Search space:

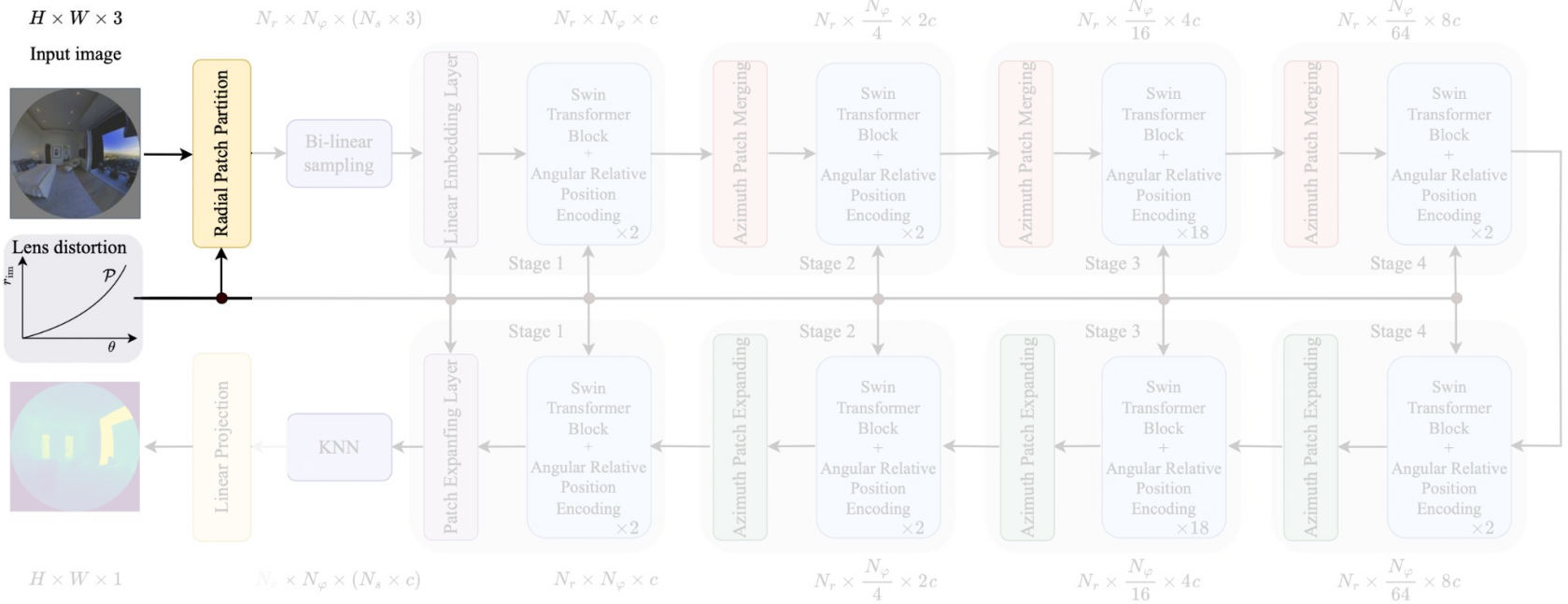
$$g(\theta)=\lambda p_n(\theta)+(1-\lambda)q_m(\theta)\,,$$

$$\text{with } p_n(\theta)=b\left(\frac{\theta}{a}\right)^n \text{ and } q_m(\theta)=1-\left(1-\frac{\theta}{a}\right)^m$$

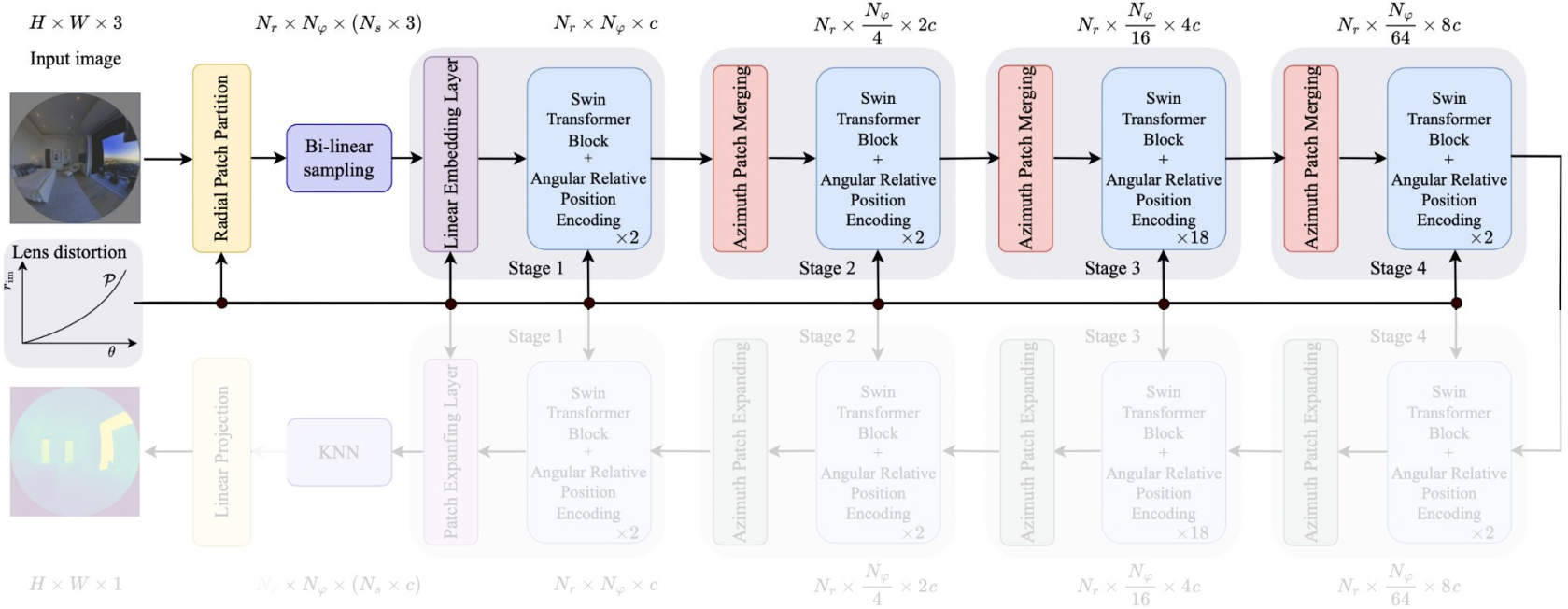
- Exhaustive search on parameters



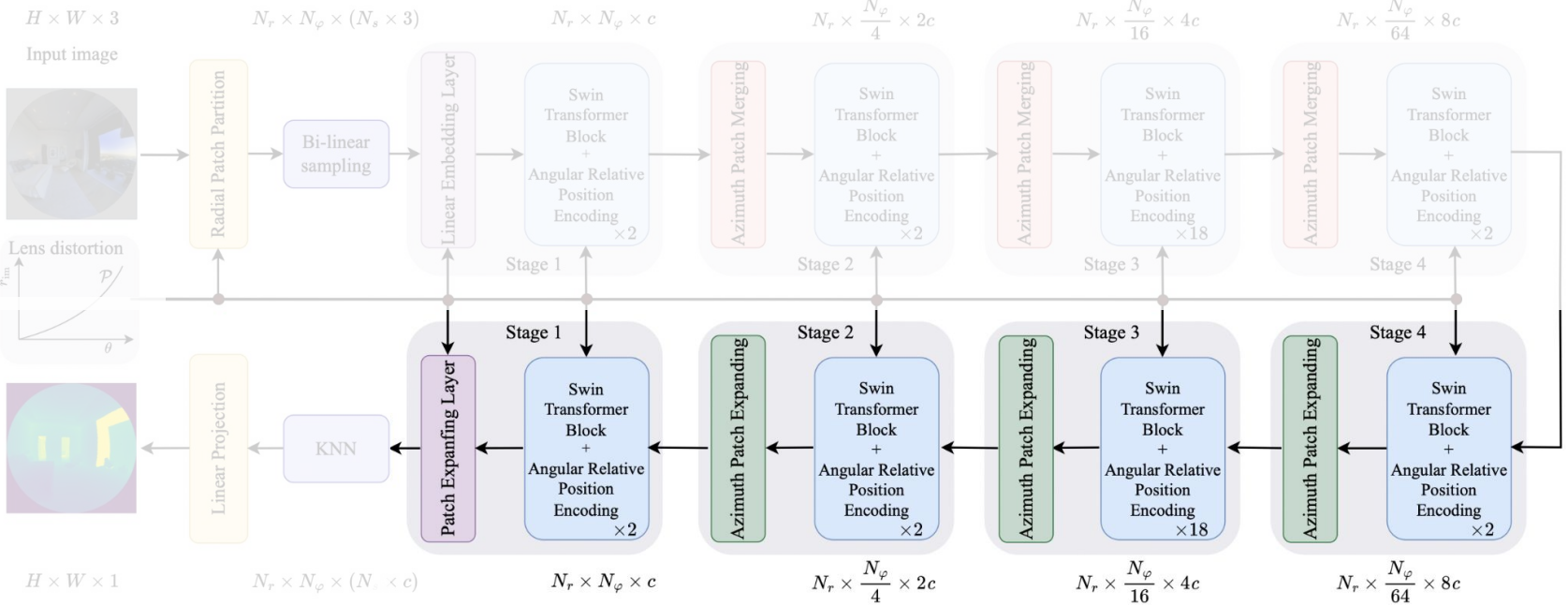
DarSwin-Unet (2025): better sampling



DarSwin-Unet (2025): encoder (DarSwin)

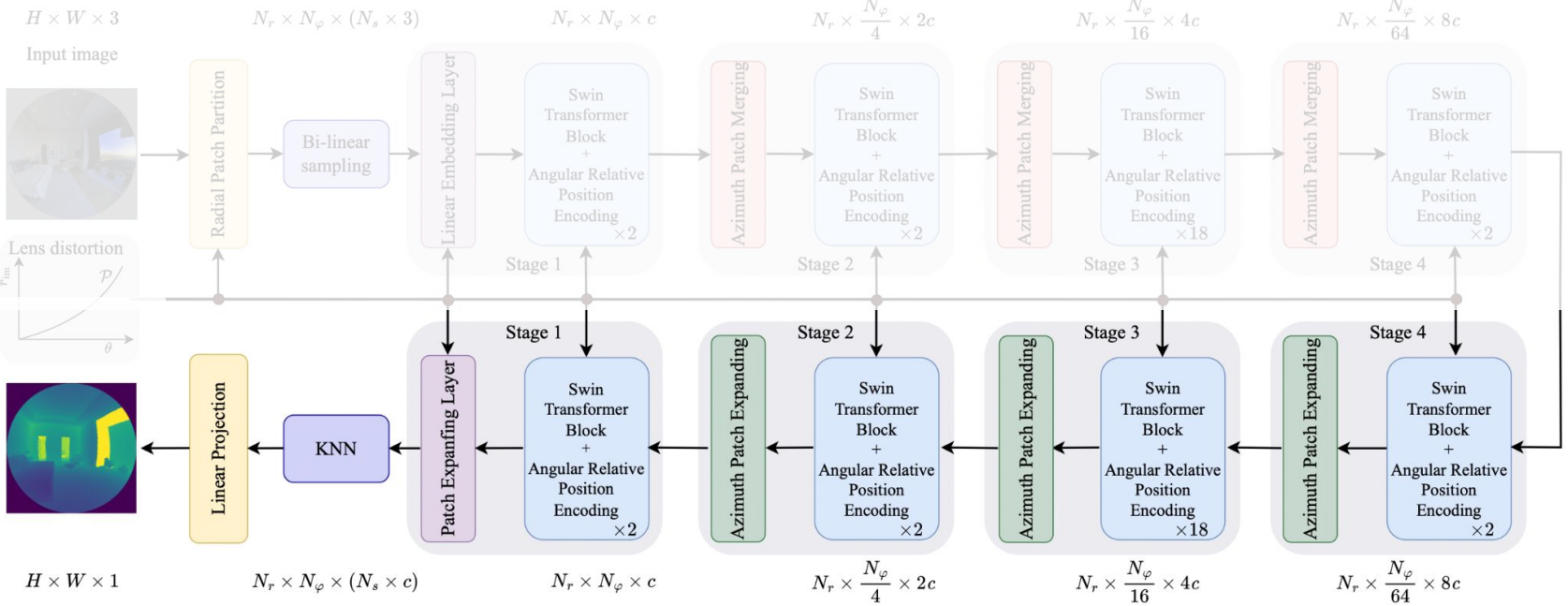


DarSwin-Unet: transformers and expanding



- MLP for the expanding layer.
- 4x expansion along the azimuth direction.

DarSwin-Unet: transformers and expanding



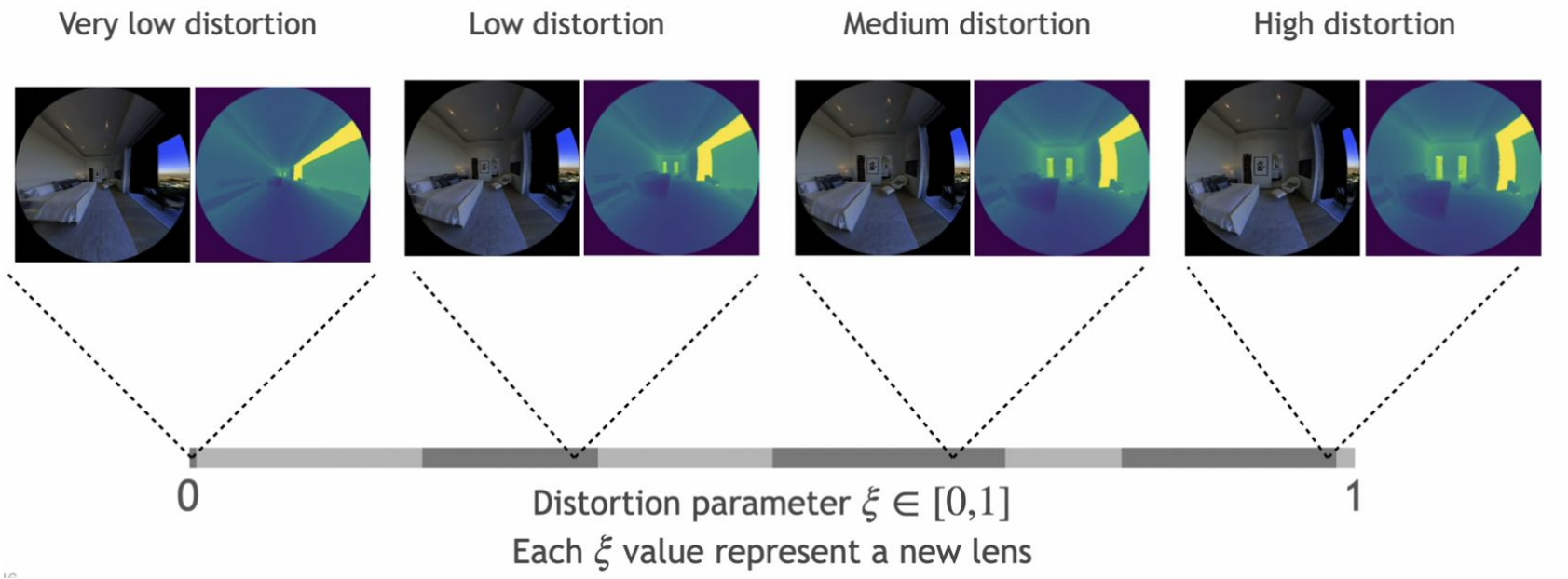
- KNN (k=4) to map to original sample resolution.
- Linear projection to obtain the result.

Training: datasets

- Matterport3D (equirectangular+depth, **indoor only**):

Training: datasets

- Simulation of varying lense distortions:



Training: datasets and loss function

- 4 sets of 9,120 training images (very low distortion, low distortion, medium distortion, and high distortion)

Training: datasets and loss function

- 4 sets of 9,120 training images (very low distortion, low distortion, medium distortion, and high distortion)

- 20 sets of 1,680 testing images ([0,1] distortion ranges subdivided in 20)

Training: datasets and loss function

- 4 sets of 9,120 training images (very low distortion, low distortion, medium distortion, and high distortion)
- 20 sets of 1,680 testing images ([0,1] distortion ranges subdivided in 20)
- Training loss: $\ell = \sqrt{\frac{1}{n}\sum_i {d_i}^2 - \frac{\lambda}{n^2}\left(\sum_i d_i\right)^2}$, where d_i is the difference between the predicted and ground-truth (log)-depth, lambda=0.85.

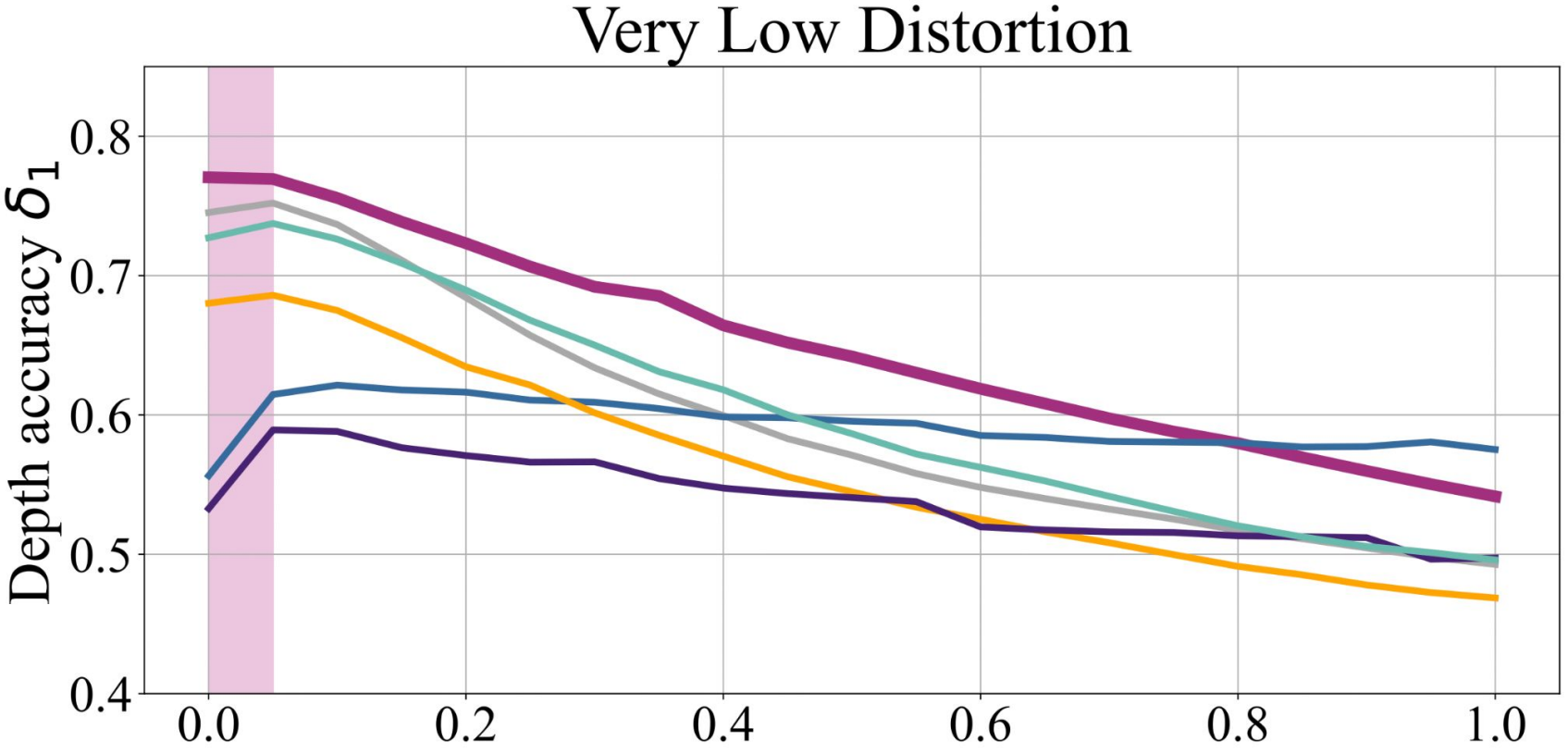
Training: datasets and loss function

- 4 sets of 9,120 training images (very low distortion, low distortion, medium distortion, and high distortion)
- 20 sets of 1,680 testing images ([0,1] distortion ranges subdivided in 20)
- Training loss: $\ell = \sqrt{\frac{1}{n}\sum_i {d_i}^2 - \frac{\lambda}{n^2}\left(\sum_i d_i\right)^2}$, where d_i is the difference between the predicted and ground-truth (log)-depth, lambda=0.85.
- Many parameters to be tuned in the network!

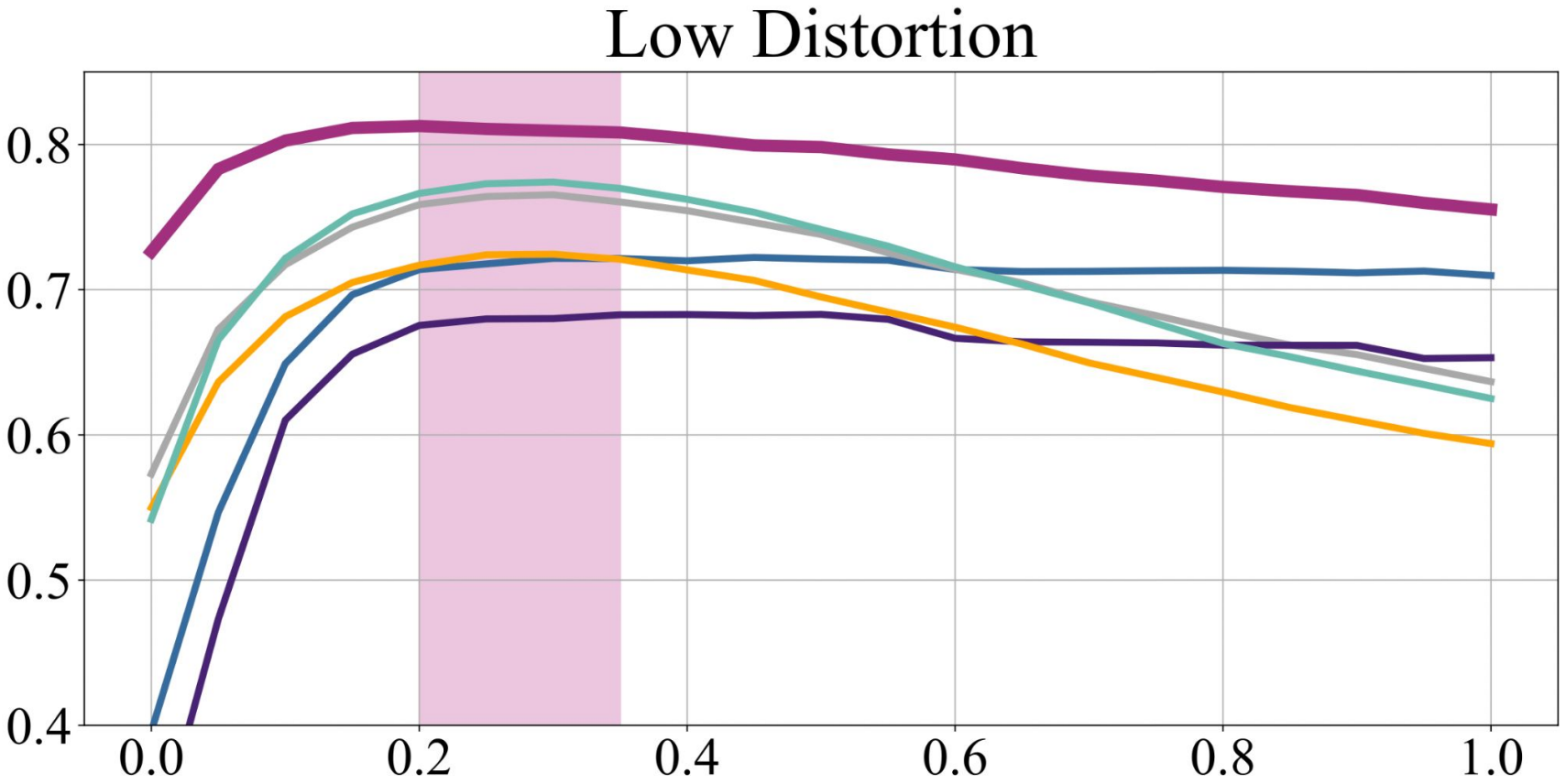
Results

- Evaluation metric: $\delta_1 = \frac{1}{|D|}|\{d \in D | \max(\frac{d^*}{d}, \frac{d}{d^*}) \leq 1.25\}|$

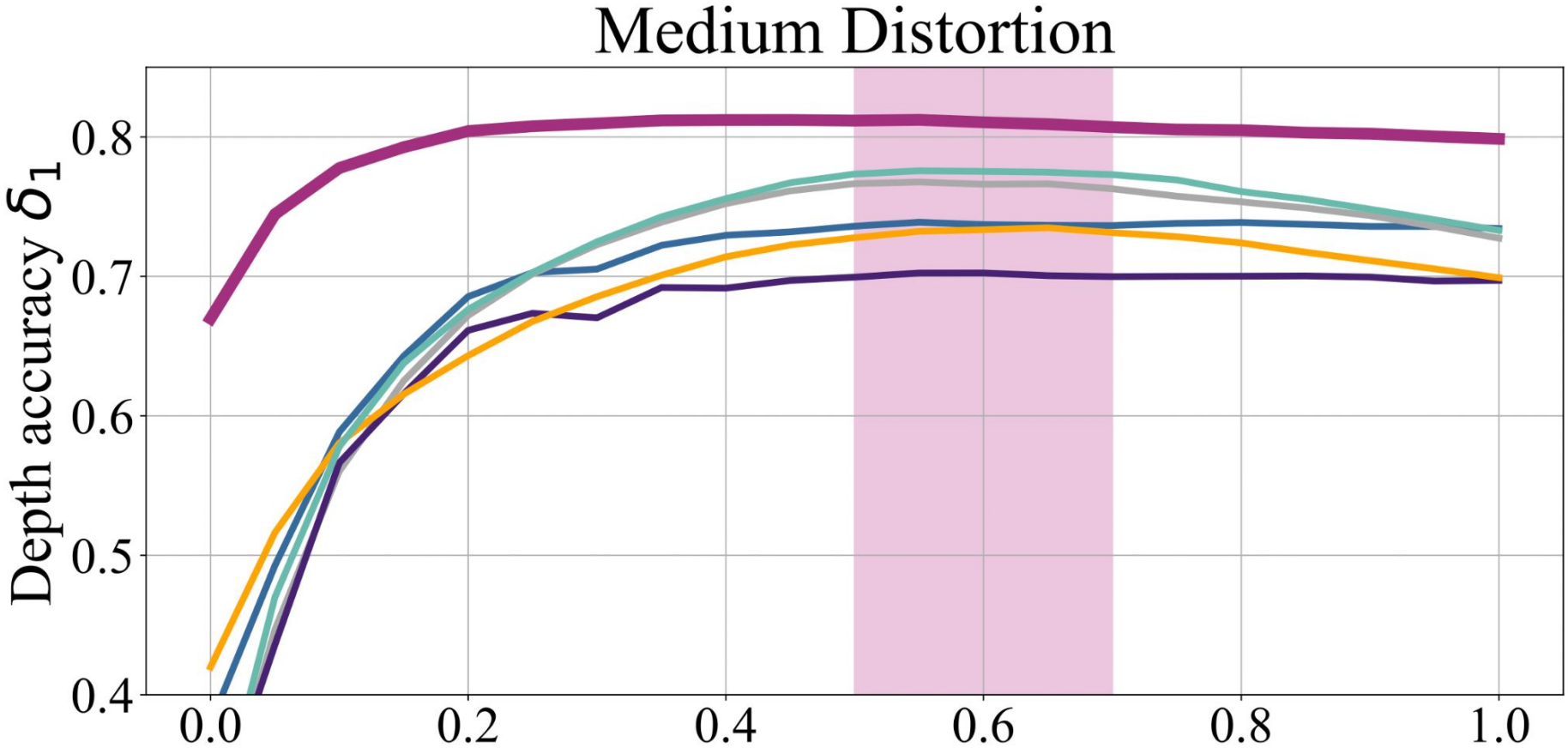
Results


Very Low Distortion
Depth accuracy δ_1
0.8
0.7
0.6
0.5
0.4
0.0
0.2
0.4
0.6
0.8
1.0

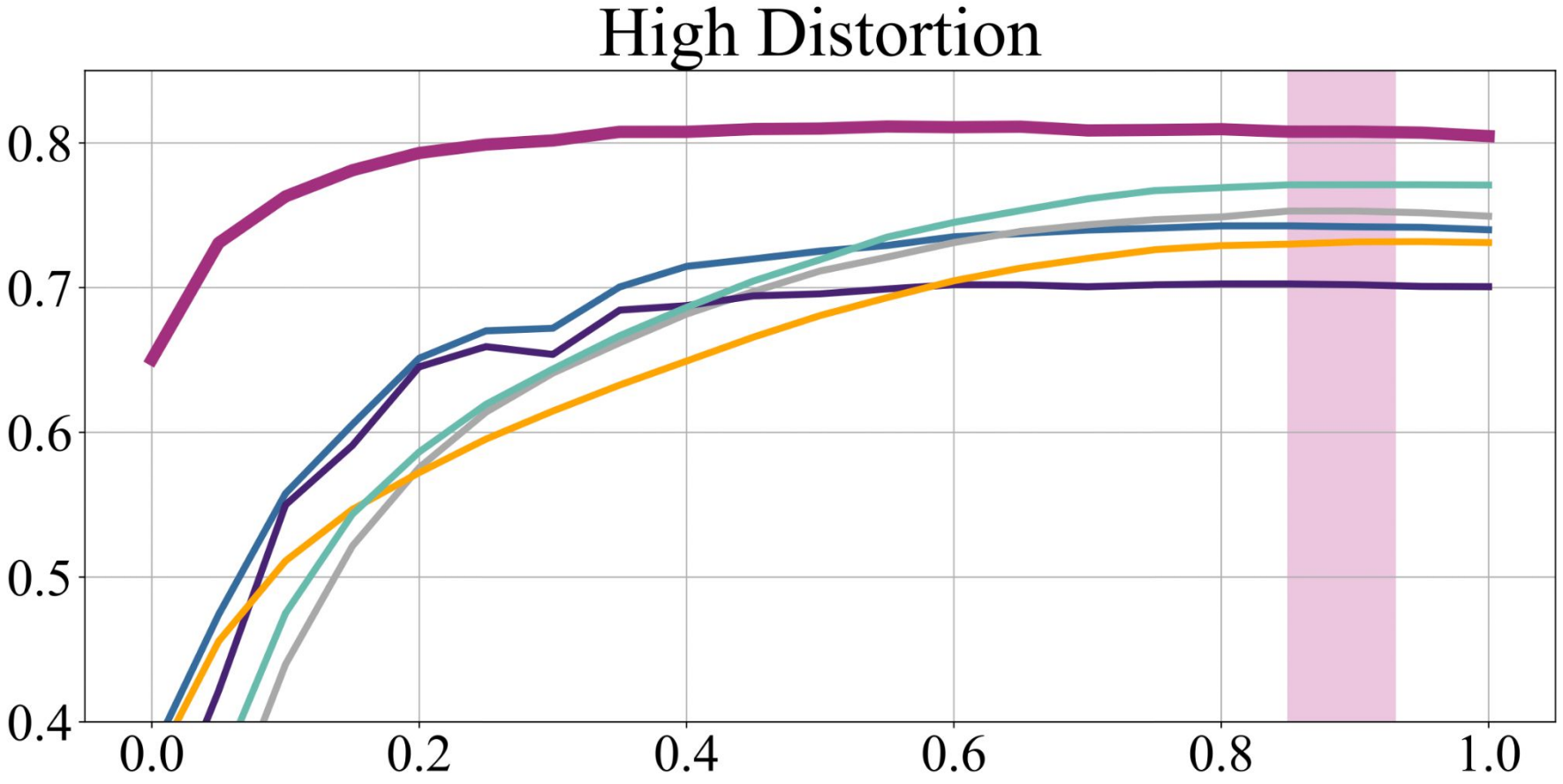
Results



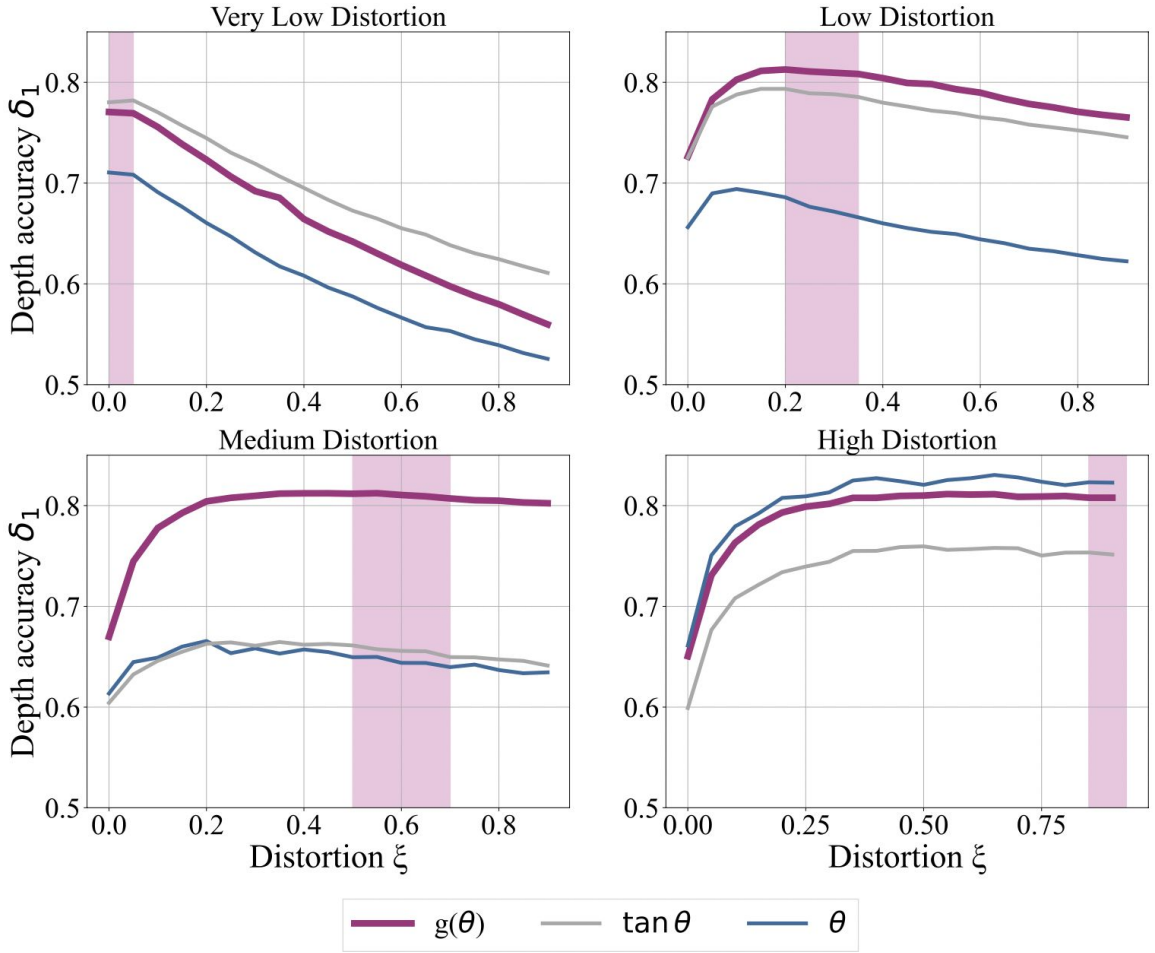
Results


Medium Distortion
Depth accuracy δ_1
0.8
0.7
0.6
0.5
0.4
0.0
0.2
0.4
0.6
0.8
1.0

Results



Results


Very Low Distortion
Low Distortion
Medium Distortion
High Distortion
Depth accuracy δ_1
Distortion ξ
g(θ)
tan θ
θ

Github repository

- https://github.com/padfoot231/DarSwin_Unet/

- Readme (1):

1. Download pre-trained swin transformer model

- [Get pre-trained model in this link for grp1] (https://hdrdb-public.s3.valeria.science/darswin-unet//home-local2/akath.extra.nobkp/DarSwin-Unet/gr1/epoch_499_1.pth)
- [Get pre-trained model in this link for grp2] (https://hdrdb-public.s3.valeria.science/darswin-unet//home-local2/akath.extra.nobkp/DarSwin-Unet/gr2/epoch_499_2.pth)
- [Get pre-trained model in this link for grp3] (https://hdrdb-public.s3.valeria.science/darswin-unet//home-local2/akath.extra.nobkp/DarSwin-Unet/gr3/epoch_499_3.pth)
- [Get pre-trained model in this link for grp4] (https://hdrdb-public.s3.valeria.science/darswin-unet//home-local2/akath.extra.nobkp/DarSwin-Unet/gr4/epoch_499_4.pth)

Github repository

De : leonardo.sacht@ufsc.br <leonardo.sacht@ufsc.br>
Date : mercredi, 22 octobre 2025 à 12:51
À : Jean-François Lalonde <jean-francois.lalonde@gel.ulaval.ca>, Lvelho <lvelho@impa.br>
Objet : DarSwin-Unet repository with broken links

Dear Jean-François Lalonde,

My name is Leonardo Sacht, I'm a professor/researcher at UFSC, Brazil. I am collaborating with Professor Luiz Velho (cced) on a project involving wide-angle images.

We were impressed by your paper "DarSwin-Unet: Distortion Aware Encoder Decoder Architecture" and wanted to try its code, but the Github repository
https://github.com/padfoot231/DarSwin_Unet
has broken links on its Readme instructions. More specifically,
https://hdrdb-public.s3.valeria.science/darswin-unet//home-local2/akath.extra.nobkp/DarSwin-Unet/gr1/epoch_499_1.pth
and similar links on the "1. Download pre-trained swin transformer model" section are broken.

Is it possible for you to fix this problem or provide alternative links to the pre-trained models?

Thank you,
Leonardo.

Github repository

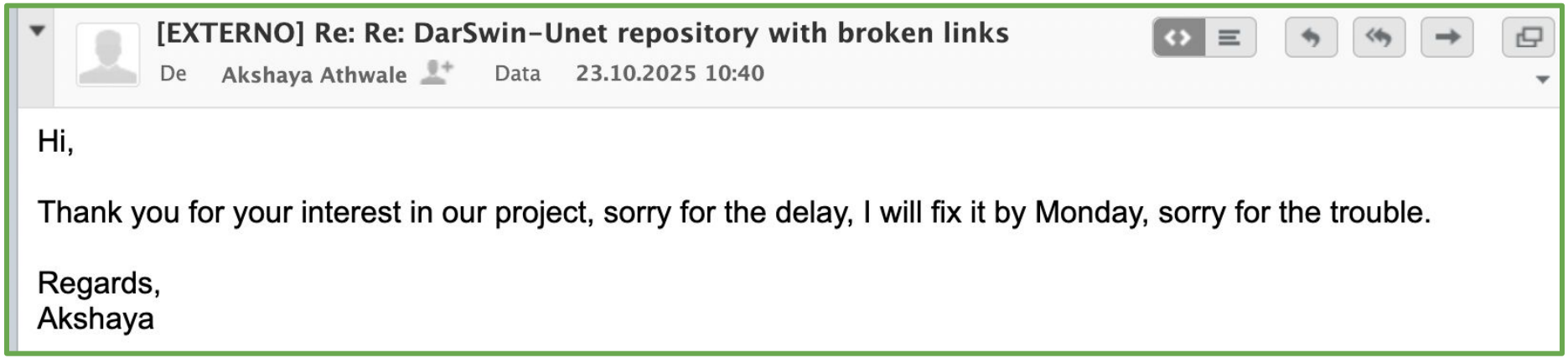
From: leonardo.sacht@ufsc.br <leonardo.sacht@ufsc.br>
Date: Wednesday, October 22, 2025 at 6:00 PM
To: Jean-François Lalonde <jean-francois.lalonde@gel.ulaval.ca>
Cc: Lvelho <lvelho@impa.br>, Akshaya Athwale <akshaya.athwale.1@ulaval.ca>
Subject: Re: Re: DarSwin-Unet repository with broken links

Thank you, Jean-François!

Em 22.10.2025 14:51, Jean-François Lalonde escreveu:

> Hi Leonardo,
>
> Thank you so much for letting us know. Actually, the links on the webpage are not broken, they're simply the wrong ones 😉
>
> It should be : https://hdrdb-public.s3.valeria.science/darswin-unet/epoch_499_1.pth
>
> @leonardo.sacht@ufsc.br : in the meantime you can download all the checkpoints by removing the /home-local2/akath.extra.nobkp/DarSwin-Unet/gr1/ from the links.
>
> @Akshaya Athwale: can you fix all the links in the readme please?
>
> Cheers,
> Jean-Francois

Github repository



[EXTERNO] Re: Re: DarSwin-Unet repository with broken links

De Akshaya Athwale Data 23.10.2025 10:40

Hi,

Thank you for your interest in our project, sorry for the delay, I will fix it by Monday, sorry for the trouble.

Regards,
Akshaya

Github repository

[EXTERNO] Re: Re: DarSwin-Unet repository with broken links

De Akshaya Athwale Data 23.10.2025 10:40

Hi,

Thank you for your interest in our project, sorry for the delay, I will fix it by Monday, sorry for the trouble.

Regards,
Akshaya

1. Download pre-trained swin transformer model

- [Get pre-trained model in this link for grp1] (https://hdrdb-public.s3.valeria.science/darswin-unet//home-local2/akath.extra.nobkp/DarSwin-Unet/gr1/epoch_499_1.pth)
- [Get pre-trained model in this link for grp2] (https://hdrdb-public.s3.valeria.science/darswin-unet//home-local2/akath.extra.nobkp/DarSwin-Unet/gr2/epoch_499_2.pth)
- [Get pre-trained model in this link for grp3] (https://hdrdb-public.s3.valeria.science/darswin-unet//home-local2/akath.extra.nobkp/DarSwin-Unet/gr3/epoch_499_3.pth)
- [Get pre-trained model in this link for grp4] (https://hdrdb-public.s3.valeria.science/darswin-unet//home-local2/akath.extra.nobkp/DarSwin-Unet/gr4/epoch_499_4.pth)

Github repository

- Readme (2):

2. Environment

- Please prepare an environment with python=3.7, and then use the command "pip install -r requirements.txt" for the dependencies.

- requirements.txt:

```
5     attrs==23.1.0
6     certifi @ file:///croot/certifi_1671487769961/work/certifi
7     cffi==1.15.1
```

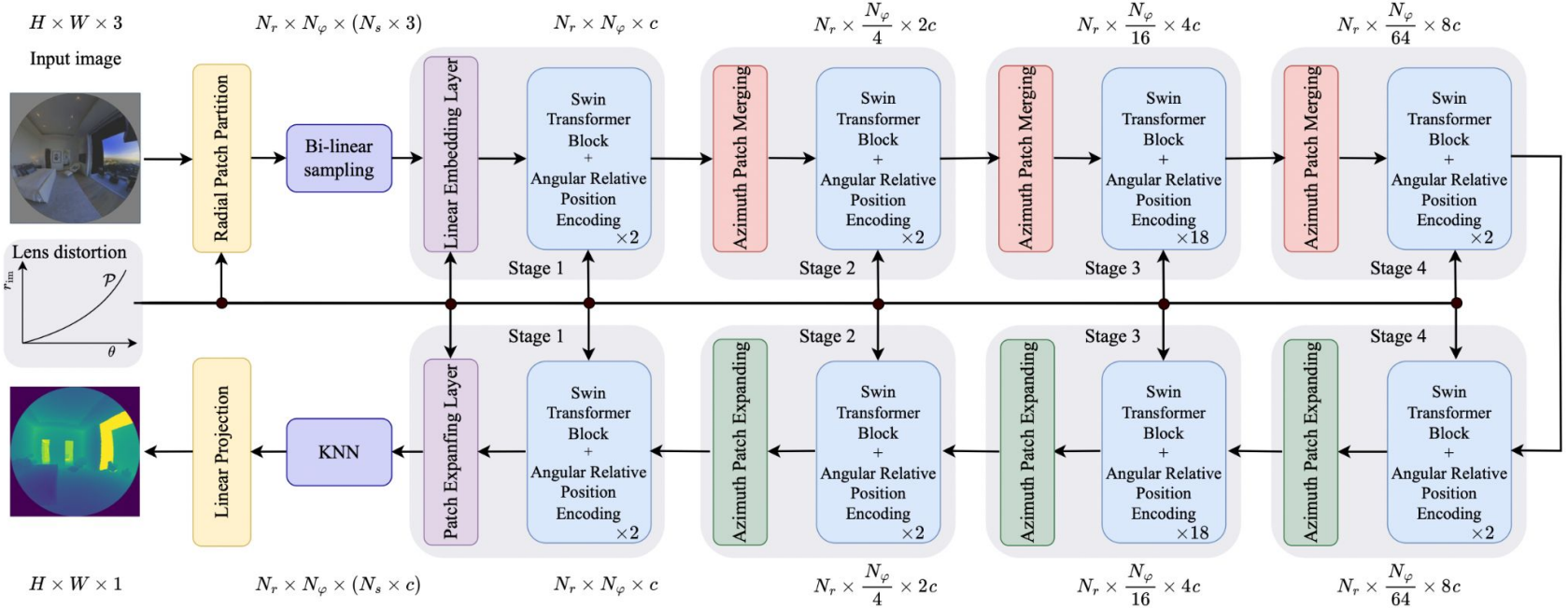
```
22    flake8==5.0.4
23    flit_core @ file:///opt/conda/conda-bld/flit-core_1644941570762/work/source/flit_core
24    fonttools==4.38.0
```

Github repository

- And running the code I get:

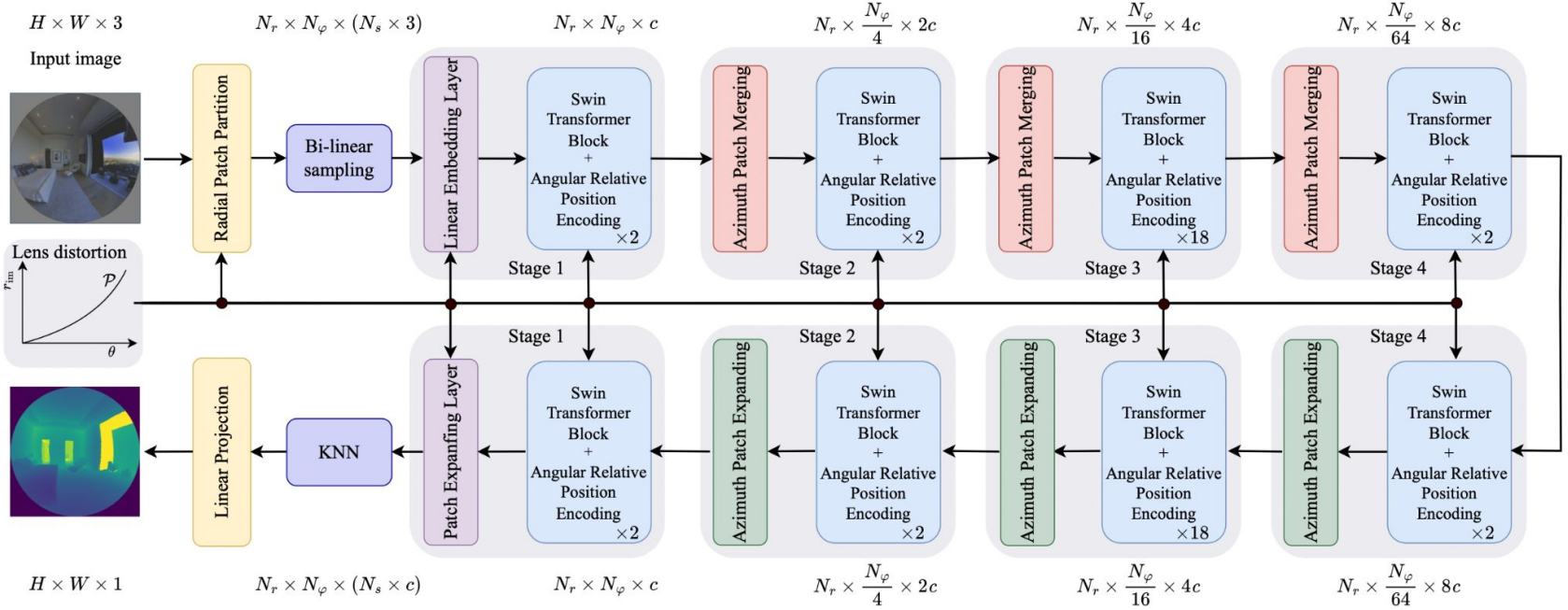
```
  File "/Users/leonardosacht/DarSwin_Unet/networks/radial_swin_transformer_unet_tan.py", line 636, in __init__
    self.blocks = nn.ModuleList([
                               ^
  File "/Users/leonardosacht/DarSwin_Unet/networks/radial_swin_transformer_unet_tan.py", line 637, in <listcomp>
    SwinTransformerBlock(dim=dim, patch_size=patch_size, input_resolution=input_resolution,
  File "/Users/leonardosacht/DarSwin_Unet/networks/radial_swin_transformer_unet_tan.py", line 352, in __init__
    self.attn = WindowAttention(
                ^^^^^^^^^^^^^^^^
  File "/Users/leonardosacht/DarSwin_Unet/networks/radial_swin_transformer_unet_tan.py", line 226, in __init__
    radius = (relative_coords[:, :, 0]).cuda(cuda_id)
              ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/opt/homebrew/lib/python3.11/site-packages/torch/cuda/__init__.py", line 403, in _lazy_init
    raise AssertionError("Torch not compiled with CUDA enabled")
AssertionError: Torch not compiled with CUDA enabled
leonardosacht@Leonardos-Air DarSwin_Unet %
```

Conclusions about DarSwin-Unet



- Positive: showed that distortion must be taken into account
- Negative: combination of many techniques, parameters, ...

Conclusions about DarSwin-Unet



- Positive: showed that distortion must be taken into account
- Negative: combination of many techniques, parameters, ...

Thank you!

Backup slides...

DarSwin-Unet: motivations

- Transformers instead of CNNs: CNNs are more suitable for perspective images (translational equivariance).

- Swin-Transformers instead of regular transformers: regular transformers are more suitable for text processing.

- DarSwin instead of Swin: Swin doesn't take into account distortions.

- DarSwin-Unet instead of DarSwin: DarSwin is encoder-only and works for non-pixel-level tasks (e.g., classification).