Optical neural network

What's the difference between computational imaging and optical computing?

MA-Yuanzheng
Dec 16th, 2021

Outline

- **Analogy of ONN**
- ONN implementation
- A glimpse
- A optics-inspired design

History

inline hologram

- Inline: The Reference wave is **co-axis** with object wave.
- Hologram: Two wave interfere beyond the object and the interference pattern recorded **at some distance.**
- Plane wave **v.s.** spherical wave (Gabor hologram)

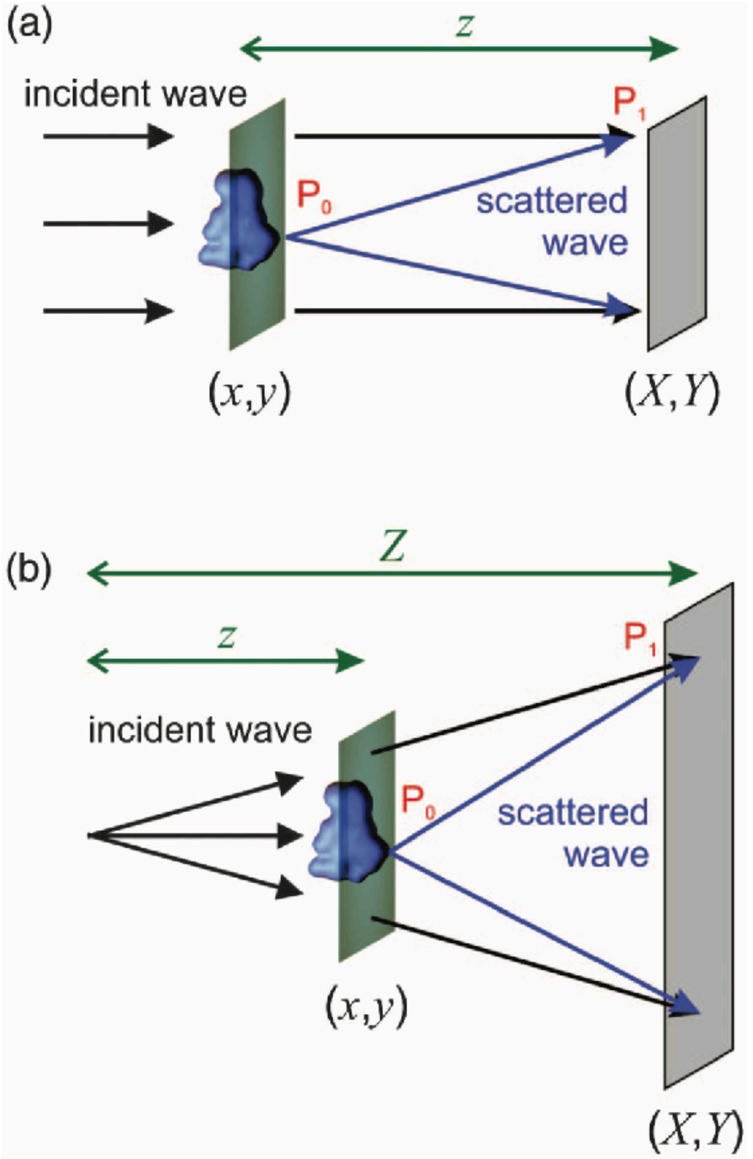


Fig. 1: In-line holography schemes realized with a **(a)** plane wave **(b)** a sperial wave

Practical algorithms for simulation and reconstruction of digital in-line holograms (2015, Hans-Werner Fink)

History

inline hologram

- Forwarding

object plane's incident optical wave & exit wave :

$U_{exit\,wave}(x,y) = U_{incident}(x,y) \cdot t(x,y)$ (1)

where $t(x,y)$ is transmission function that is

$t(x,y) = e^{-\alpha(x,y) + j\phi(x,y)}$ (absorption, phase distribution)

$= 1 + \tilde{t}(x,y)$. (2)

(2) → (1) ⇒

$U_{exit\,wave}(x,y) - U_{incident}(x,y) = U_{incident} \cdot \tilde{t}(x,y)$.

The recorded wave of image plane :

$$U_{image}(X,Y) = \frac{1}{\lambda i}\iint U_{incident}(x,y)\cdot t(x,y)\cdot \frac{e^{(ik|\vec{r}|)}}{|\vec{r}|}\,dxdy$$

$$= K\iint 1\cdot t(x,y)\cdot \frac{e^{i\frac{\pi}{\lambda z}[(x-X)^2+(y-Y)^2+z^2]}}{|(x-X)^2+(y-Y)^2+z^2|^{\frac{1}{2}}}$$

$|z| \gg \rho^2/\lambda$. $= K\iint 1\cdot t(x,y)\cdot \frac{e^{i\frac{\pi}{\lambda z}[(x-X)^2+(y-Y)^2]}}{|z|}\,dxdy$

$$= \frac{K}{z}\iint t\cdot(x,y)\cdot e^{i\frac{\pi}{\lambda z}[(x-X)^2+(y-Y)^2]}\,dxdy$$

$$= \frac{K}{z}\cdot t(X,Y) \otimes s(X,Y)$$ (⊗ : conv.)

- Back-propagation

The recorded hologram ⇒ $H(X,Y) = |U_{image}(X,Y)|^2$ (conjugate)

$= |R|^2 + |O|^2 + R^*O + RO^*$

normalization ⇒ $H_{norm} = 1 + \frac{|O|^2}{|R|^2} + \frac{R^*O}{|R|^2} + \frac{RO^*}{|R|^2}$ ($\frac{|O|^2}{|R|^2} \to 0$)

$\Rightarrow H_0 = H_{norm} - 1 \doteq \frac{R^*O + RO^*}{|R|^2}$ ($bg = 0$)

* H_0 · hanning to reduce hologram edge effect

also Fresnel-Kirchoff ⇒ $U(x,y) = K\iint R(X,Y)H_0(X,Y)\cdot e^{\frac{-ik|\vec{\Delta r}|}{|\vec{\Delta r}|}}\,dXdY$.

$R=1$ & far-field condition ⇒

$$= \frac{K}{z}\iint 1\cdot H_0\cdot e^{\frac{-ik|\vec{\Delta r}|}{|\vec{\Delta r}|}}\,dXdY$$

$$\simeq \frac{K}{z}\cdot H_0 \otimes S^*$$

$1 + U/R \Rightarrow \tilde{t} + 1 = t = e^{-\alpha}e^{-i\phi}$

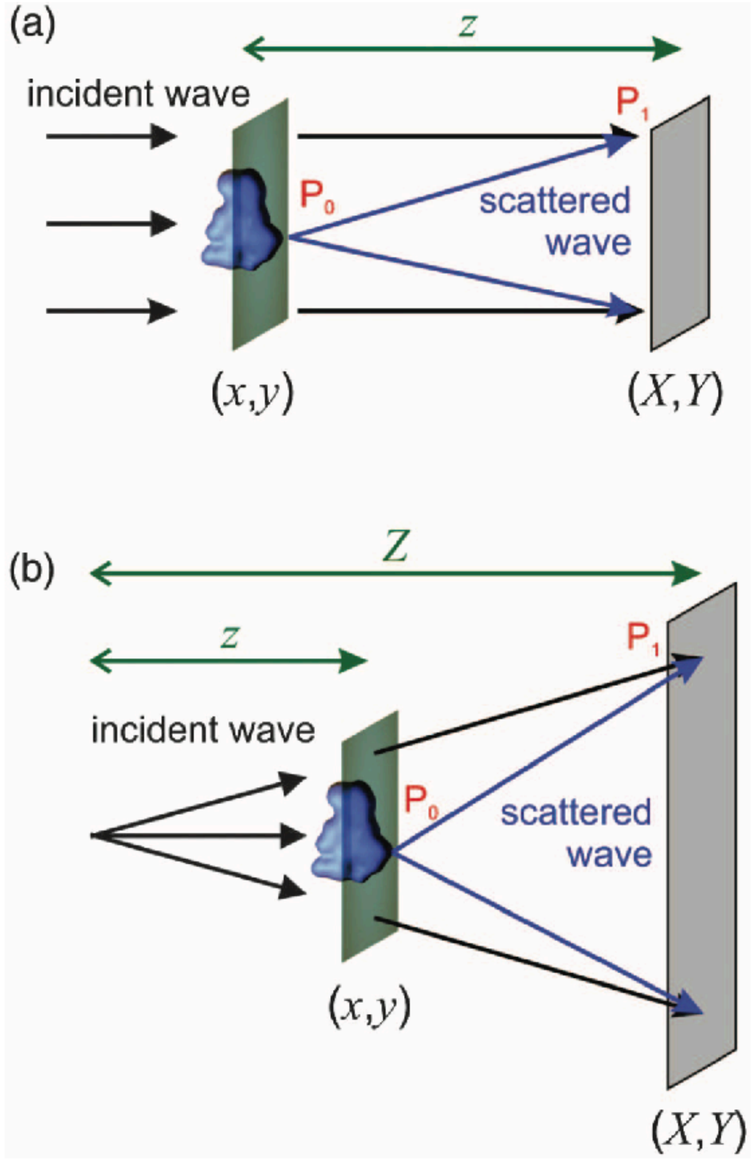


Fig. 1: In-line holography schemes realized with a **(a)** plane wave **(b)** a sperial wave

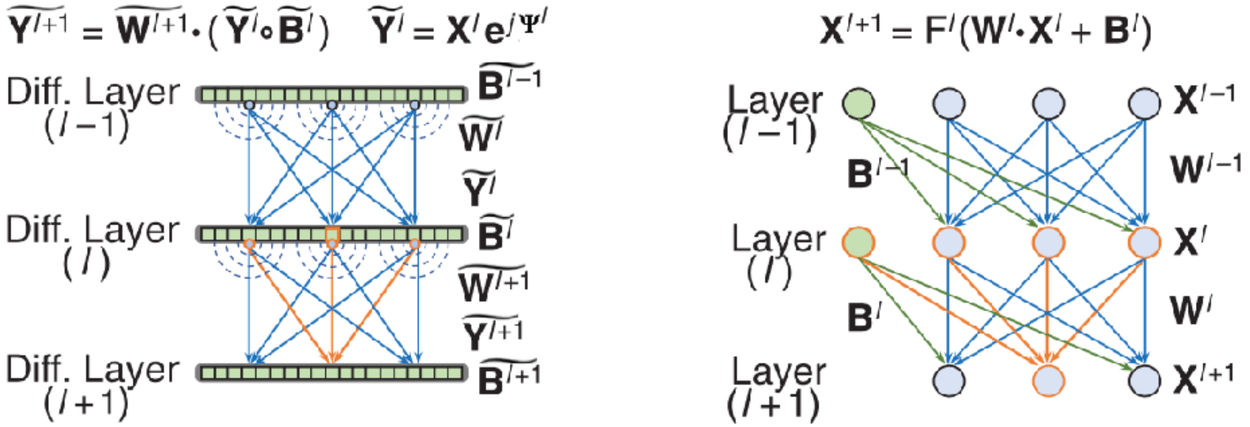
Practical algorithms for simulation and reconstruction of digital in-line holograms (2015, Hans-Werner Fink)

New Mexico
美版脱贫攻坚战

History

Optical neural network

- A optical implementation of an ANN
- Hopfield neural netowrk
- Dispersive medium for optical analog of the neural network Applied Optics, Marcus S. Cohen
- Diffractive neural network



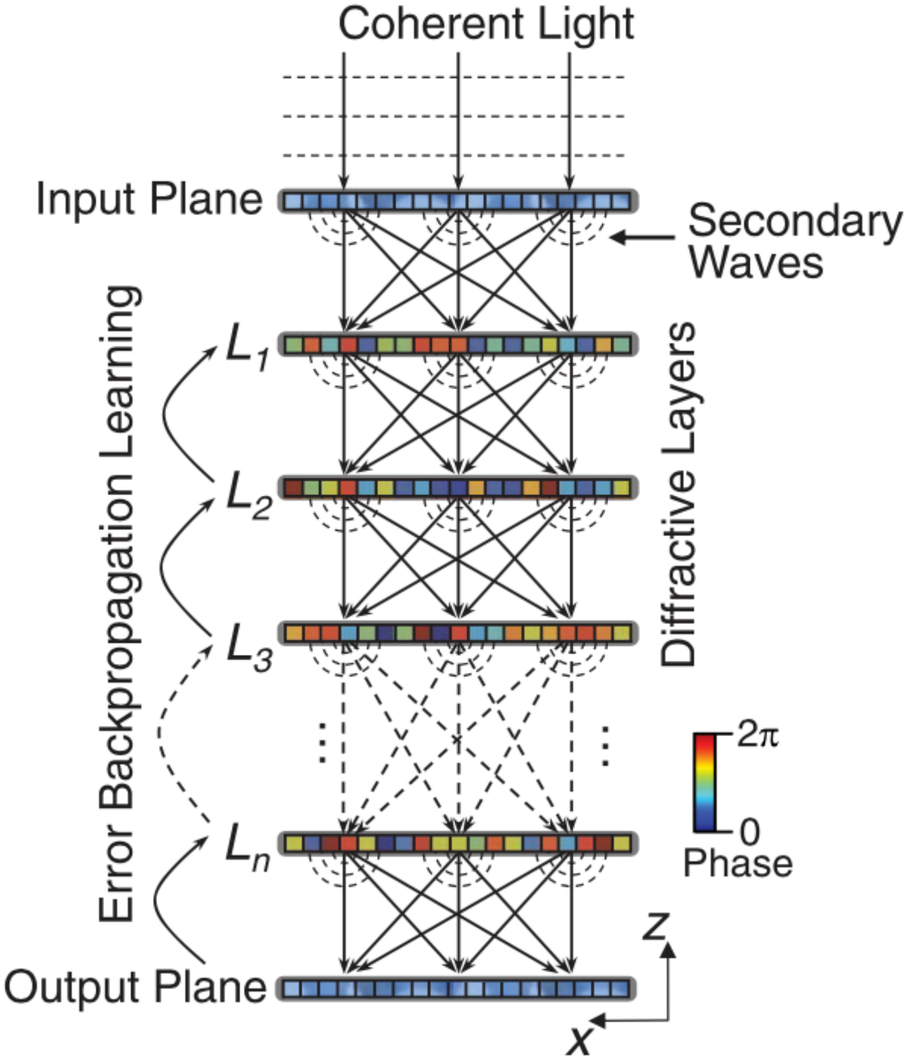


Fig. 2: Diffractive deep neural network (D2NNs)

All-optical machine learning using diffractive deep neural networks (2017, Aydogan Ozcan)

Schematic

All-optical reconstruction

1. A plane wave as illuminating source.
2. Diffraction pattern is recored.
3. Again illustrate hologram followed by diffractive neurel network.

- Rayleigh-Sommerfeld formulation of diffraction

$$u_k^l = \sum_m \frac{z_l}{(r_{mk}^l)^2}\left(\frac{1}{2\pi r_{mk}^l} + \frac{1}{j\lambda}\right)\exp\left(j\frac{2\pi r_{mk}^l}{\lambda}\right)v_m^{l-1}$$

$$r_{mk}^l = \sqrt{(x_k - x_m)^2 + (y_k - y_m)^2 + z_l^2}$$

λ is the wavelength of the optical wave;
z is the axial l distance between layers l − 1 and l;
(x_m, y_m) and (x_k, y_k) are the transverse coordinates of feature m of layer l − 1 and feature k of layer l;
v^0 and u^{L+1} are the optical fields at the input and the output fields-of-view of the diffractive network.

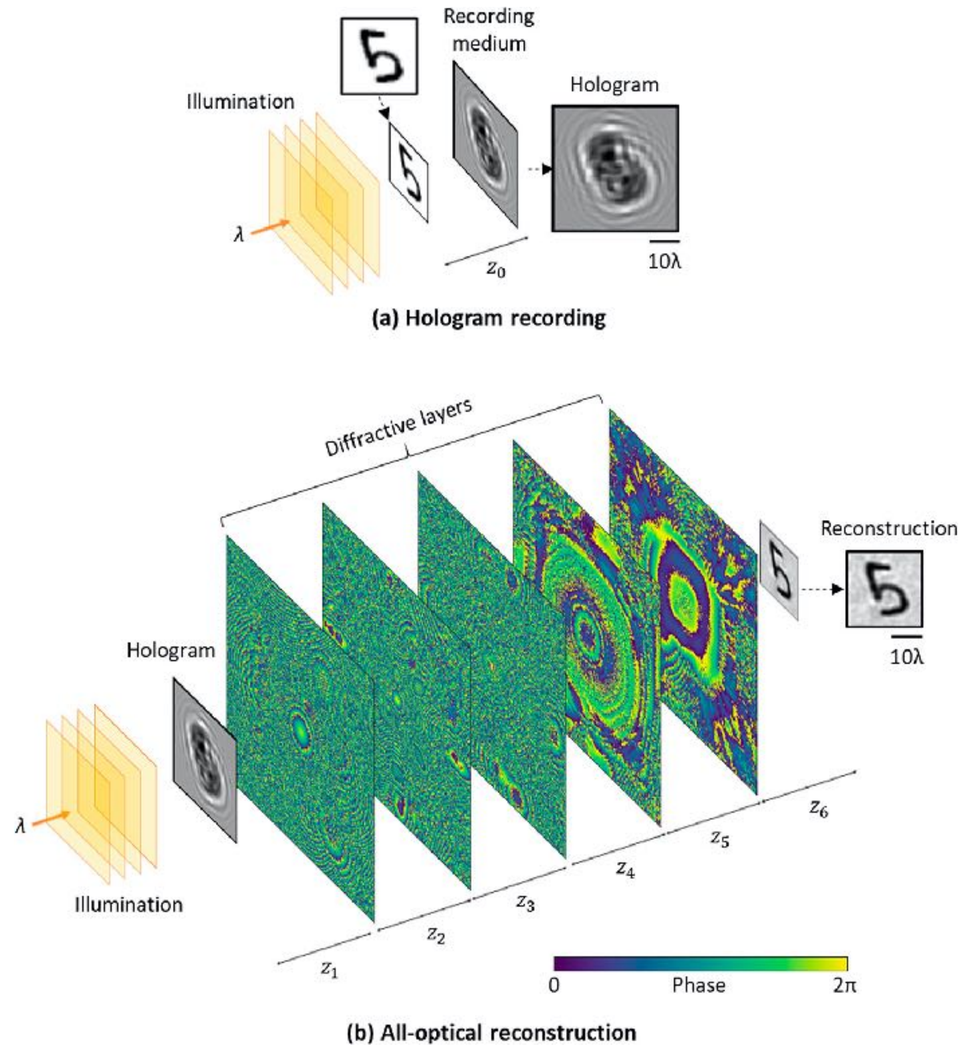


Fig. 3: Hologram recording and reconstruction scheme for the proposed all-optical holograhpic reconstruction method

Computer-Free, All-Optical Reconstruction of Holograms Using Diffractive Networks (2021, Aydogan Ozcan)

Schematic

All-optical reconstruction

- Complex-valued transmittance of pixel k of layer l is denoted with t_k^l
- A thin diffractive layer
- Thus ===> $v_k^l = u_k^l \cdot t_k^l = \{ \sum_m w_{km}^l v_m^{l-1} \} \cdot t_k^l$
- where $w_{km}^l = \frac{z_l}{(r_{mk}^l)^2} \left(\frac{1}{2\pi r_{mk}^l} + \frac{1}{j\lambda} \right) \exp\left(j \frac{2\pi r_{mk}^l}{\lambda} \right)$

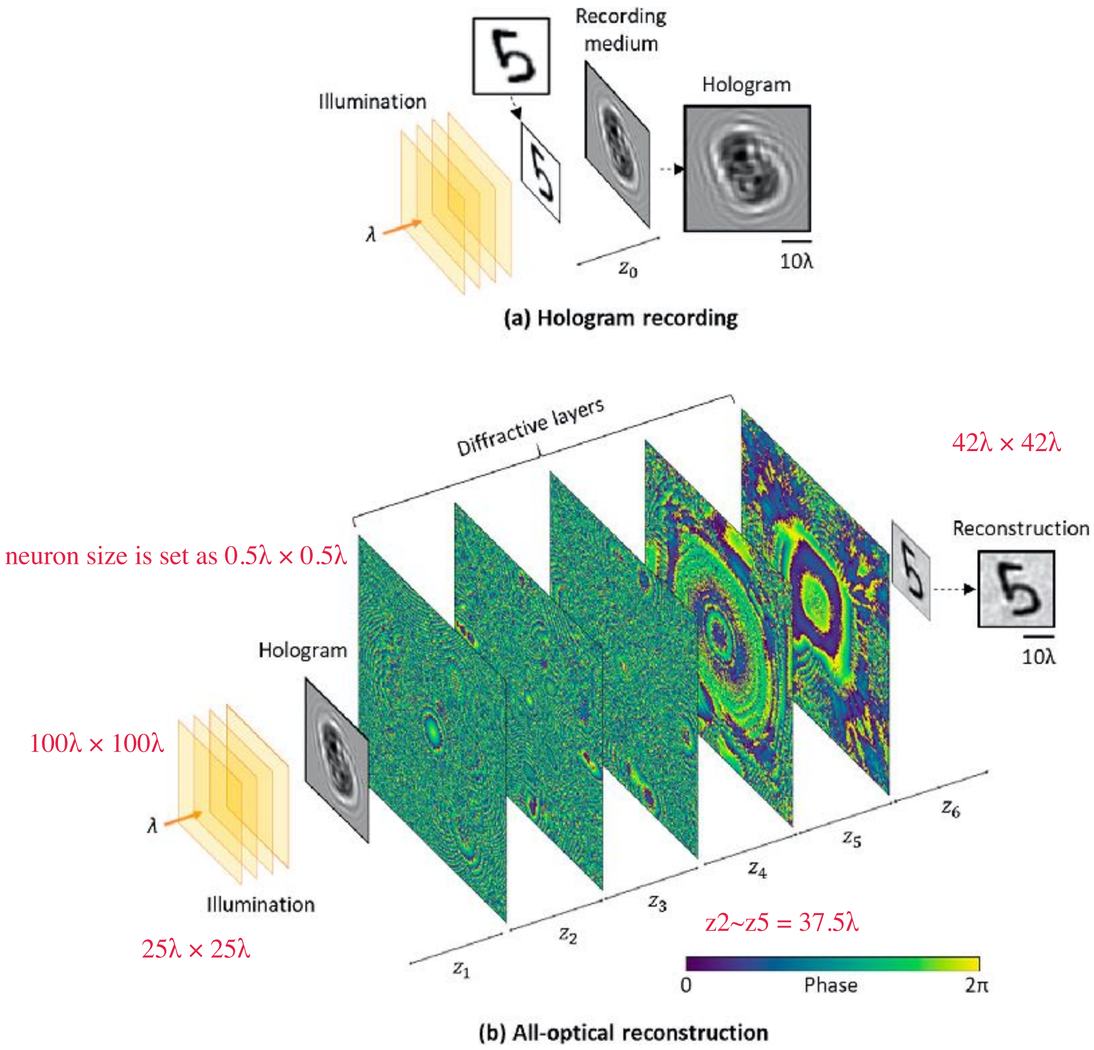


Fig. 3: Hologram recording and reconstruction scheme for the proposed all-optical holograhpic reconstruction method (λ = 600 nm)

Computer-Free, All-Optical Reconstruction of Holograms Using Diffractive Networks (2021, Aydogan Ozcan)

Results

reconstruction with trained model

- Loss function:

$$L = L_{\text{pixel}} + 1000L_{\text{fourier}} + \eta L_{\text{efficiency}}$$

$$L_{\text{pixel}} = \frac{1}{N_p}\sum_{p=1}^{N_p} |y_p - \hat{y}_p|$$

$$L_{\text{fourier}} = \frac{1}{N_p}\sum_{p=1}^{N_p} |F\{y\}_p - F\{\hat{y}\}_p|^2$$

$$L_{\text{efficiency}} = 1 - \frac{P_{\text{out}}}{P_{\text{illum}}}$$

Where Lpixel is MSE; Lfourier is Fourier domain's MSE; Lefficiency is efficiency of light energy.

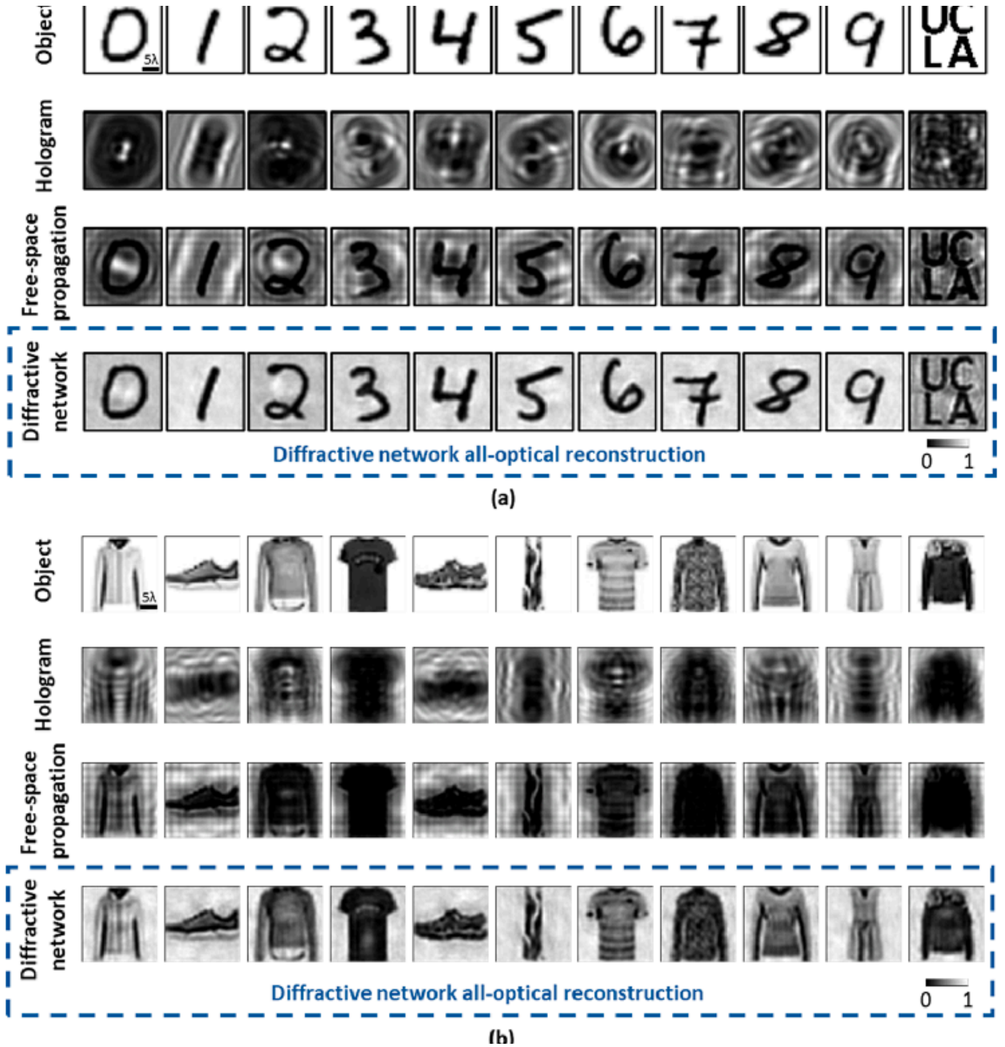


Fig. 4: Performance of all-optical hologram reconstruction using diffractive neural network (z6=30λ)

Computer-Free, All-Optical Reconstruction of Holograms Using Diffractive Networks (2021, Aydogan Ozcan)

Results

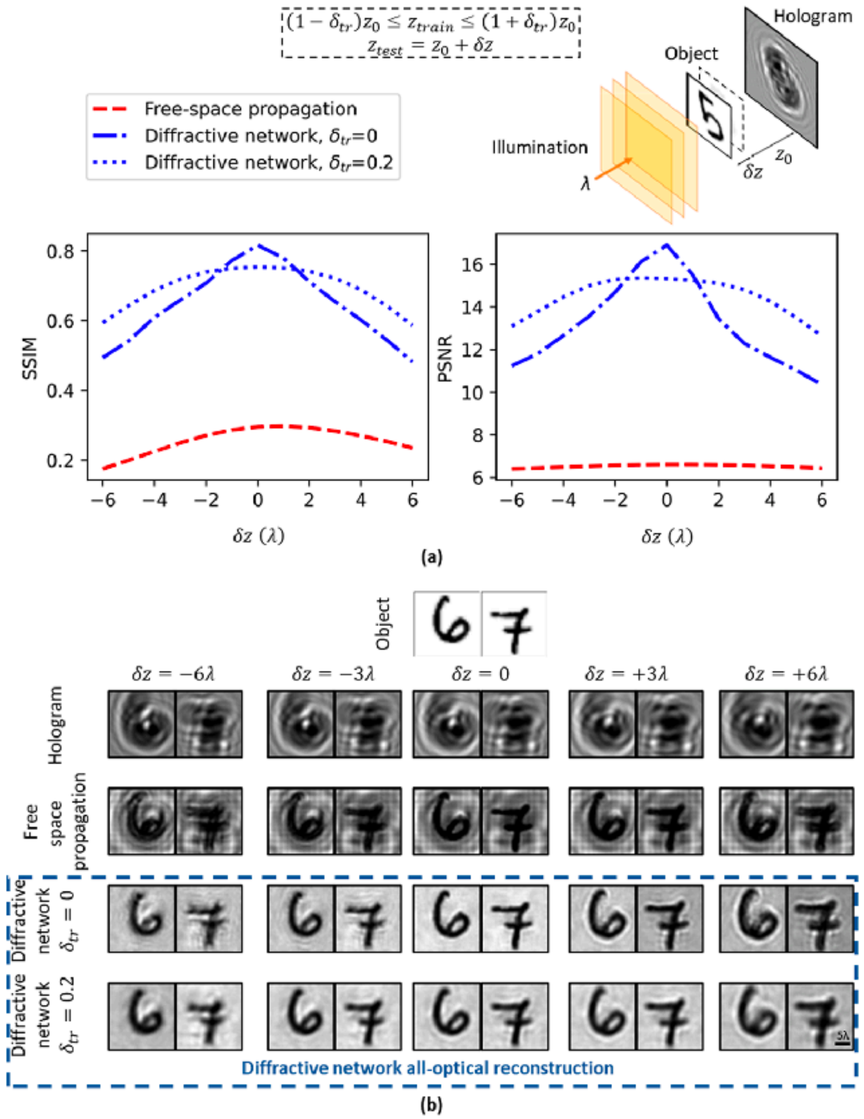


Fig. 5: Robustness of all-optical diffractive network reconstruction against **the hologram recording distance variance**.

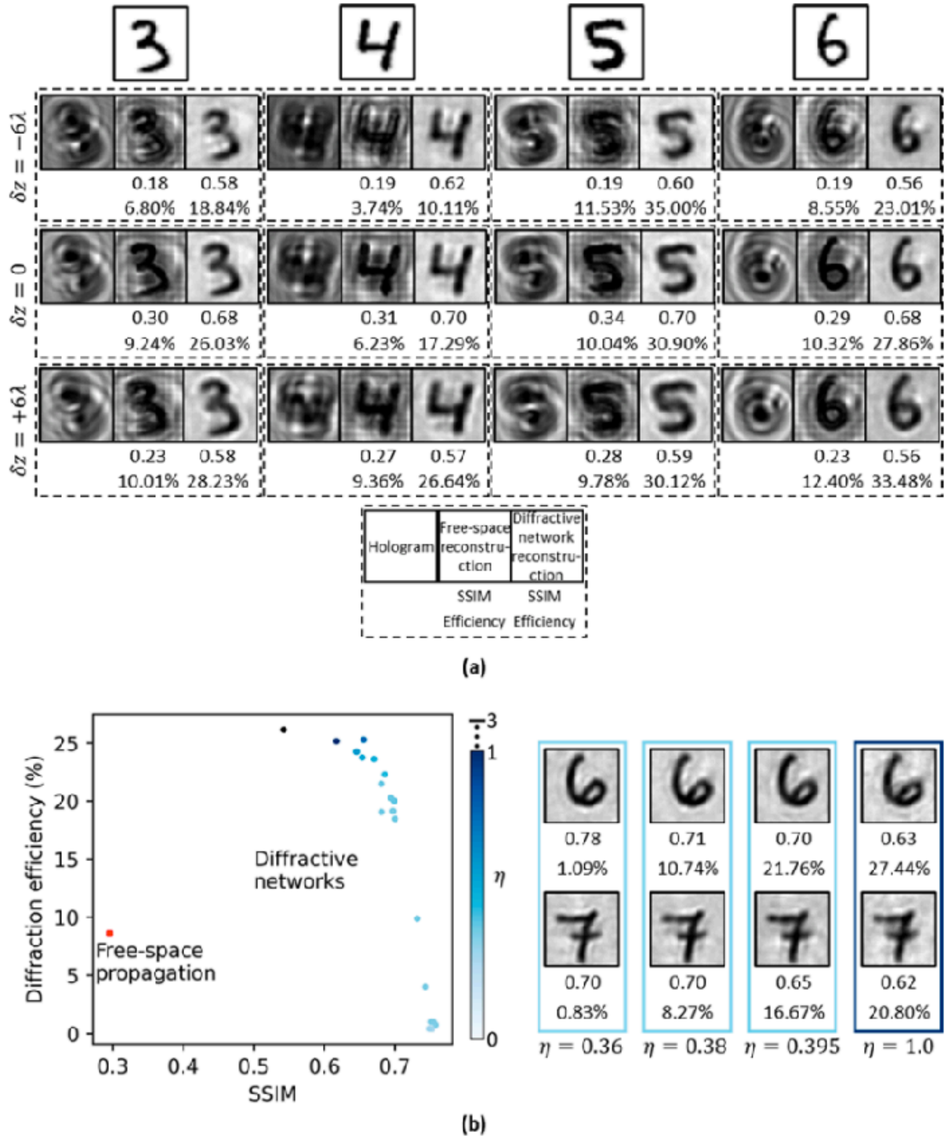


Fig. 6: Diffraction efficiency improvement of all-optical holographic reconstruction performed by diffractive network.

Conclusion

- Phase recovery & twin-image elimination based on all-optical processor (High speed).
- Diffractive layers can perform as neural (physics based connection layer).
- Method that vaccinate the model the changing of hologram recording distance.
- Rayleigh–Sommerfeld diffraction integral was computed using the Angular Spectrum method
- Tensorflow + Adam + mixed Loss GTX1080Ti + Win 10

Outline

- Analogy of ONN
- **ONN implementation**
- A glimpse
- A optics-inspired design

Schematic

The optical path of ONN

Ti-DLP9000X

Meadowlark

Andor zyla 4.2

Paraboloid representing

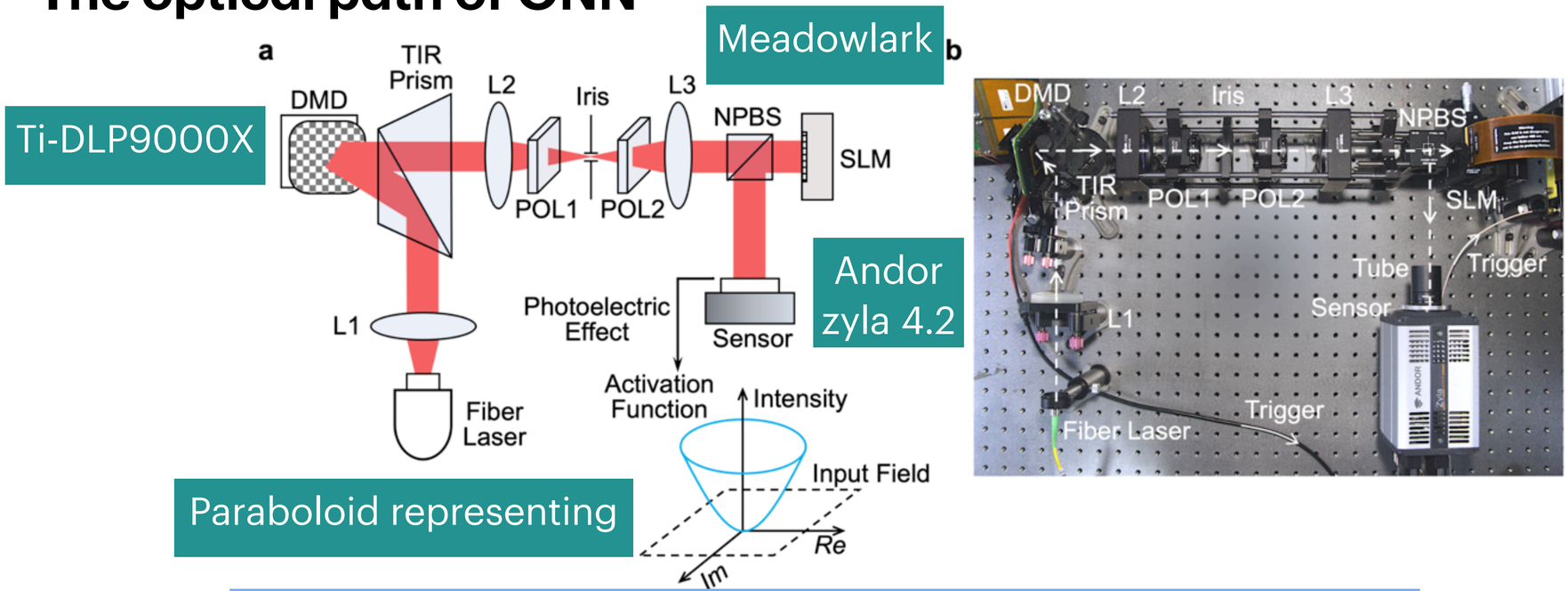


Fig. 2-1 Schematic and photo of the experimental DPU prototype

- **NPBS**: non-polarized beam splitter.
- Laser: 698nm; DMD: 149989Hz(2560 × 1600 , 1-Bit) + 7.6 μm
- SLM: 422.4Hz (1,920 × 1,152, 8-Bits) + 7.6 μm + efficiency 88% + update time of ~2.4 ms
- sCMOS: 100 f.p.s. (2048 × 2048, 16-Bits) + 6.5 μm

Large-scale neuromorphic optoelectronic computing with a reconfigurable diffractive processing unit

Dataflow

E -> O -> E

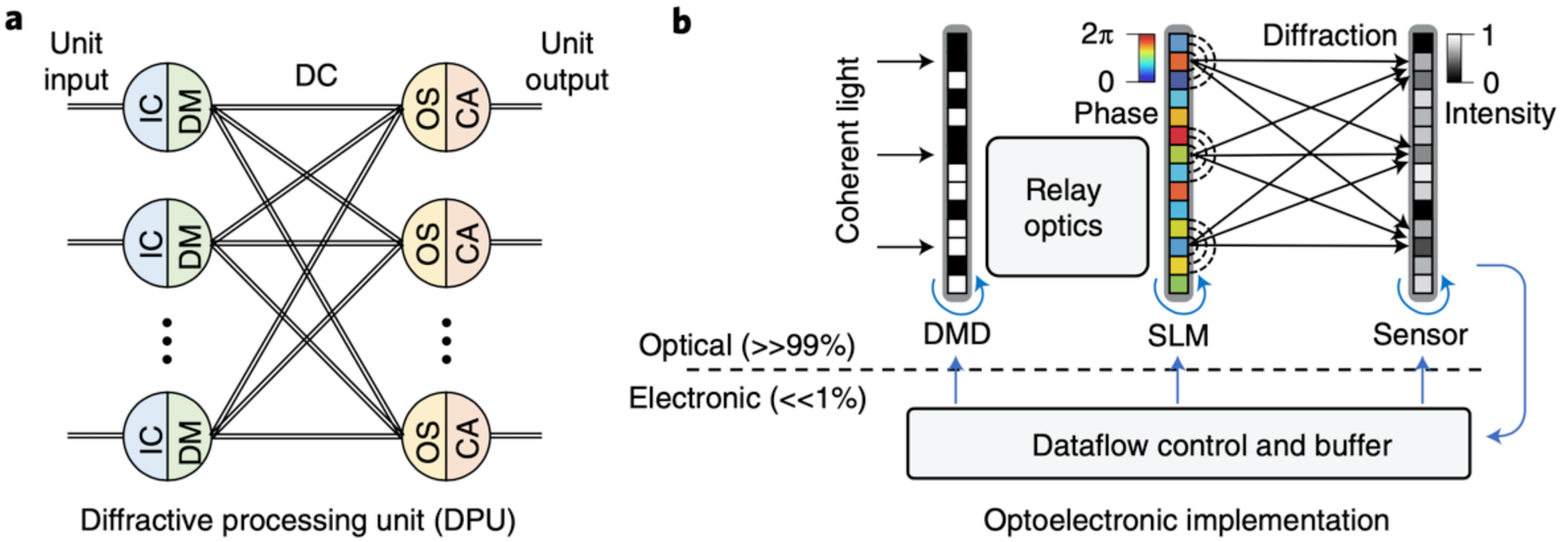


Fig. 2-2 Dataflow of perceptron-like optoelectronic computing building block

- IC: information coding; DM: Diffractive modulation; DC: Diffractive connection
- OS: optical field summation; CA: complex activation;

Large-scale neuromorphic optoelectronic computing with a reconfigurable diffractive processing unit

Network structure

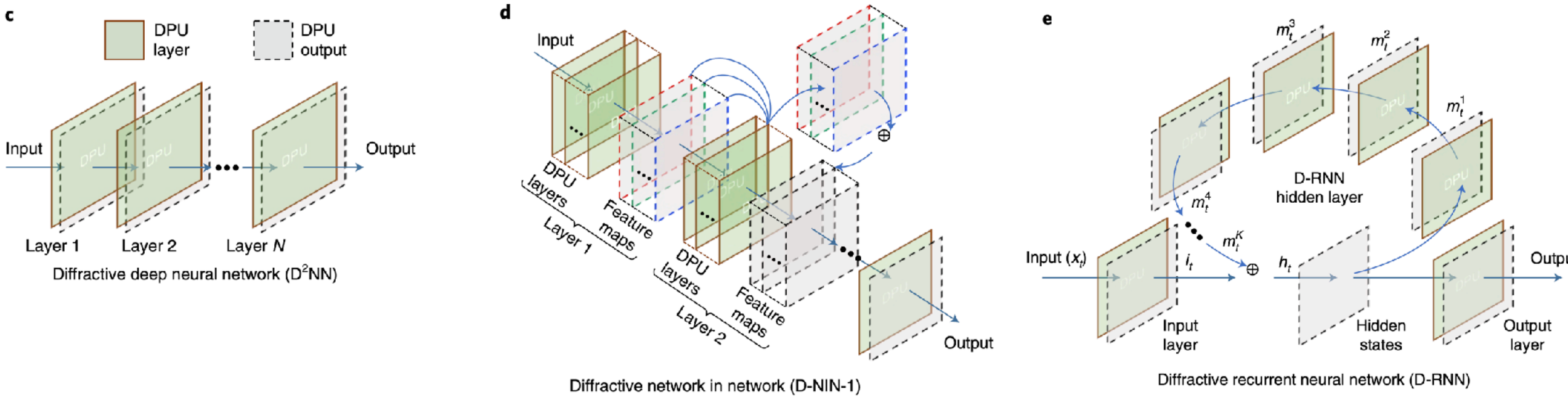


Fig. 2-3 Different DNN, that are D2NN, DNIN, D-RNN

- Single layer, multi-layers, and recurrent layer

$$\text{out}(N_i, C_{\text{out}_j}) = \text{bias}(C_{\text{out}_j}) + \sum_{k=0}^{C_{\text{in}}-1} \text{weight}(C_{\text{out}_j}, k) \star \text{input}(N_i, k)$$

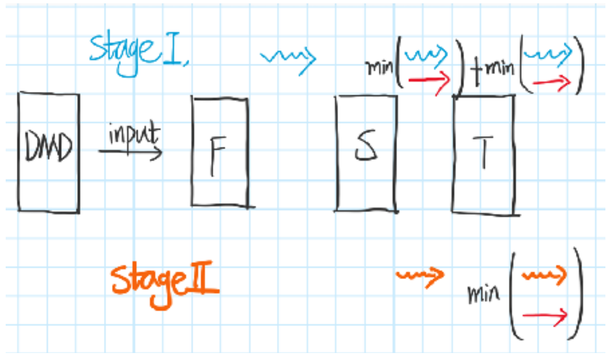
- , where (N,Cin,H,W) is the output value of the layer with input size and (N,Cout,Hout,Wout) refers to its output.

Large-scale neuromorphic optoelectronic computing with a reconfigurable diffractive processing unit

Adaptive training

inter-layer fine-turning

- **Each layer** contains transfer errors distort the wavefront connections of neurons.
- in situ v.s. in solico
- Robustness for few-shot



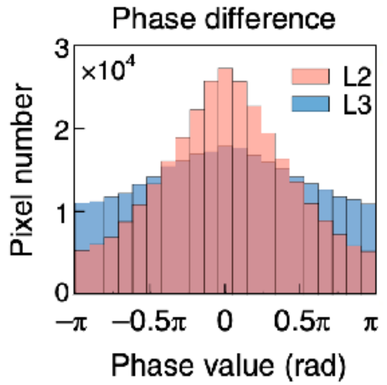


Fig. 2-5 Illustration of adaptive train & results of phase diff

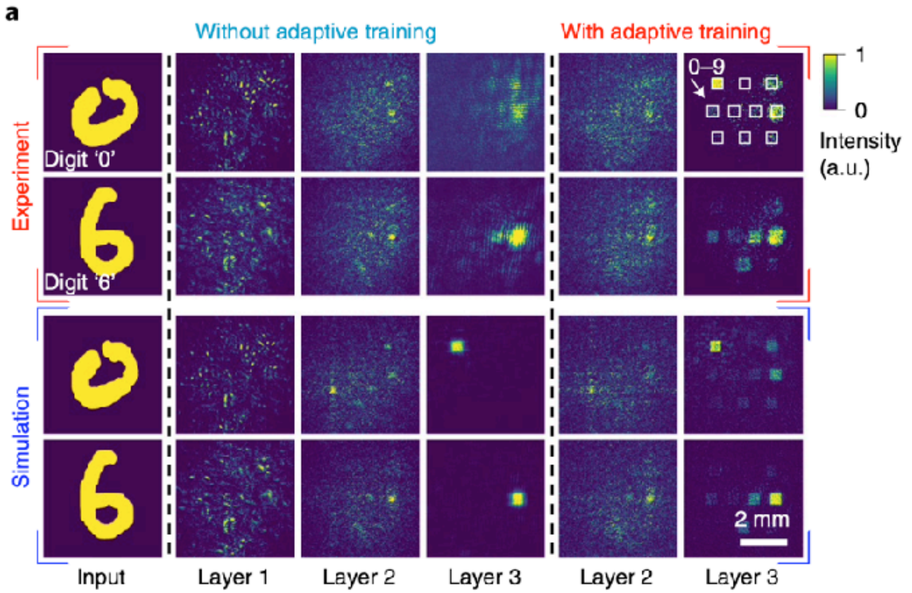


Fig. 2-4 Experimental DPU outputs of a 3-layers OE D2NN

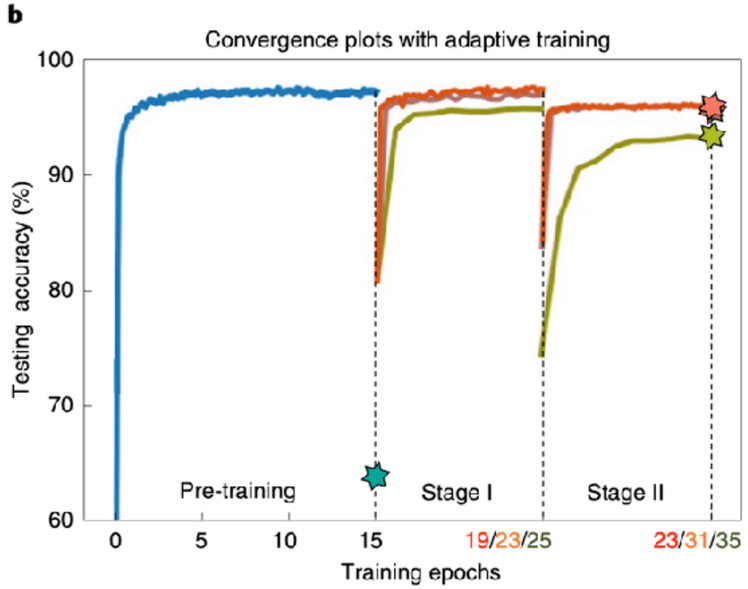


Fig. 2-6 Experimental DPU outputs of a 3-layers OE D2NN

Results

MINIST classfication

- **Simulating the multi-feature map of CNN**

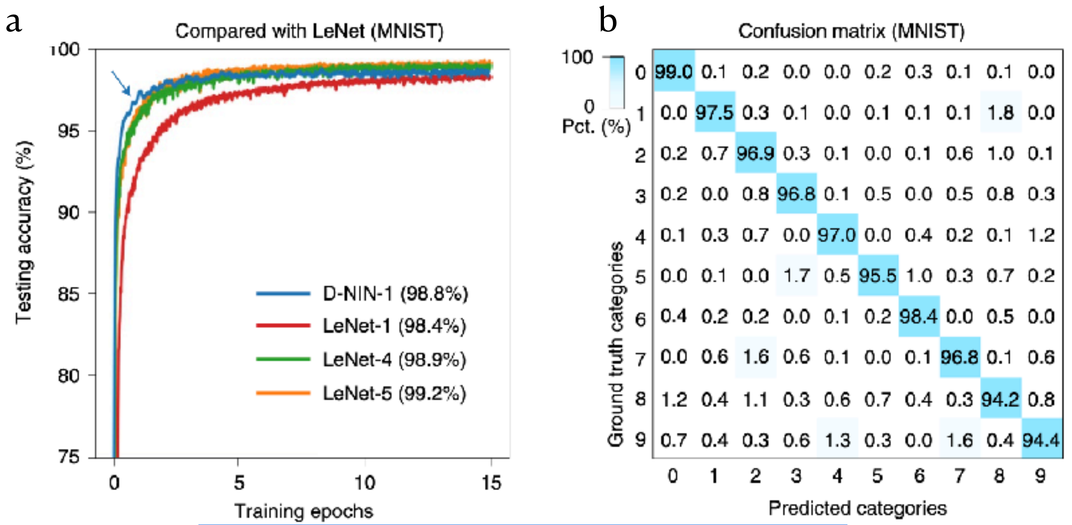


Fig. 2-6 Test results of D-NIN-1

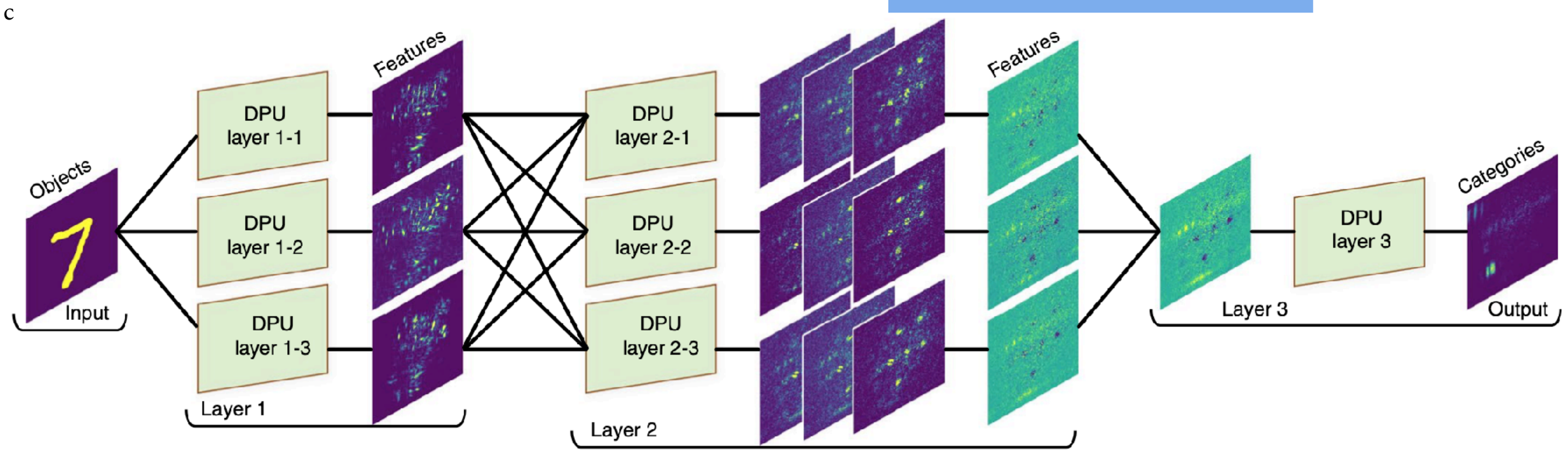


Fig. 2-7 Tensor flow detail of training with D-NIN-1

Results

optical version RNN

- Weizmann: 180*140 ppi; deinterlaced 50fps;run, walk, skip, shortly jack, jump-forward-on-two-legs, pjump, gallopsideways, wave2, wave1, or bend; 340MB.
- KTH: 160*120 ppi2; walking, jogging, running, boxing, handwaving, handclapping; 766.37MB

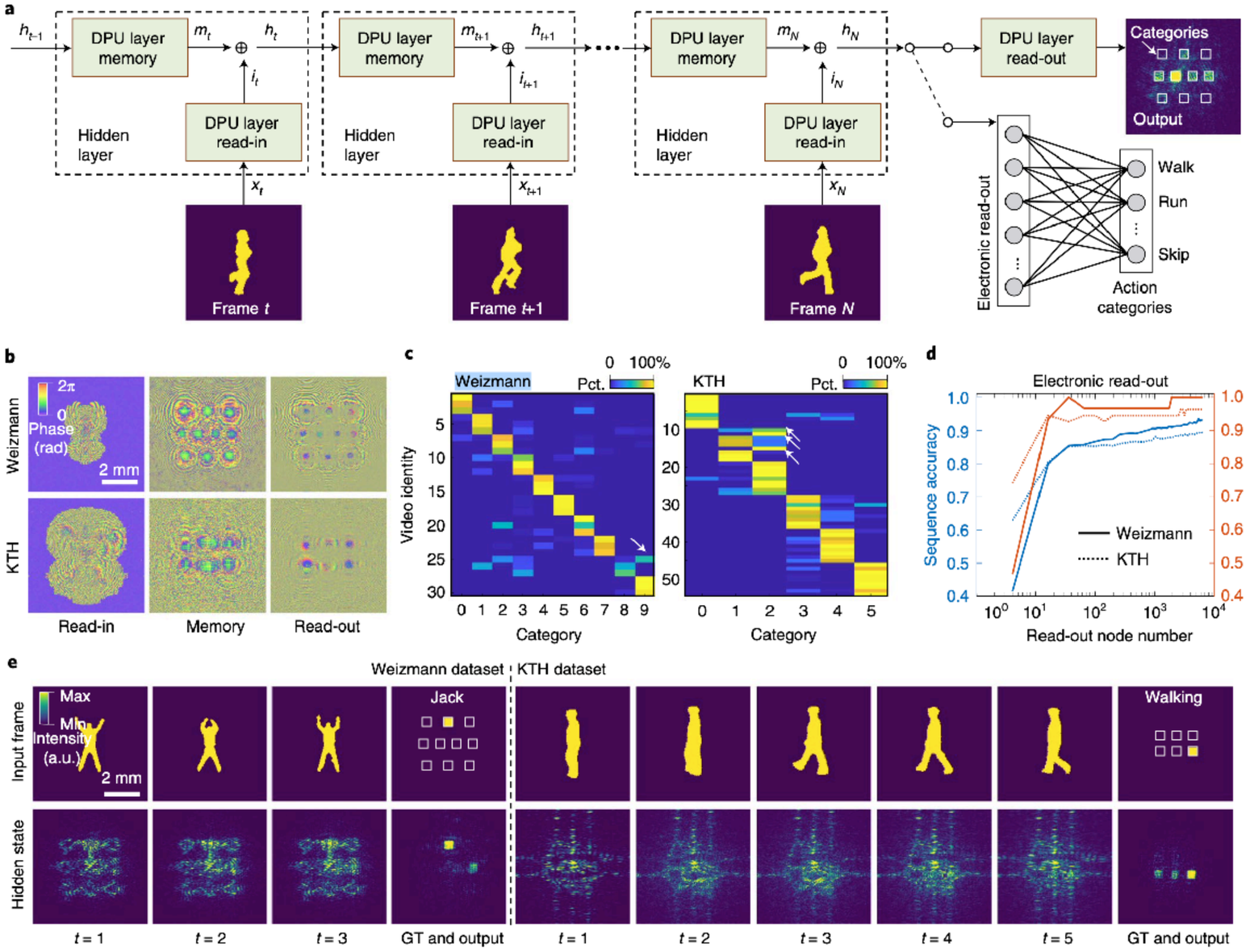


Fig. 2-8 D-RNN for human actions recognition

Conclusion

- Still a **data-driven** training.
- Speed of training is determined by **speed of DMD and sCMOS and SLM**
- Data preprocessing is important (interpolation & featuring)
- Good inducing may come from **multi-task learning** & **widely varying dataset**.
- Classification is benefited from **average pooling** of the network output.
- softmax cross-entropy := softmax loss = $E = -\sum_{j=1}^{T} y_i log(\underbrace{\frac{e^{a_i}}{\sum_{k=1}^{T} e^{a_k}}}_{\text{softmax}})$, where yi is ith class in array, e^{ai} is value of **exp** (neuron output).

Outline

- Analogy of ONN
- ONN implementation
- **A glimpse**
- A optics-inspired design

Prior knowledge

What is Vision Transformer (ViT)?

- Attention is all you need (Transformer)
- An image is worth 16*16 words (ViT)
- BERT->RNN -» LSTM -» Transformer (to // tn)

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1...m}\}$;
Parameters to be learned: γ, β
Output: $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \text{// mini-batch mean}$$

$$\sigma^2_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_{\mathcal{B}})^2 \qquad \text{// mini-batch variance}$$

$$\widehat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma^2_{\mathcal{B}} + \epsilon}} \qquad \text{// normalize}$$

$$y_i \leftarrow \gamma\widehat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i) \qquad \text{// scale and shift}$$

where x as input & y is output

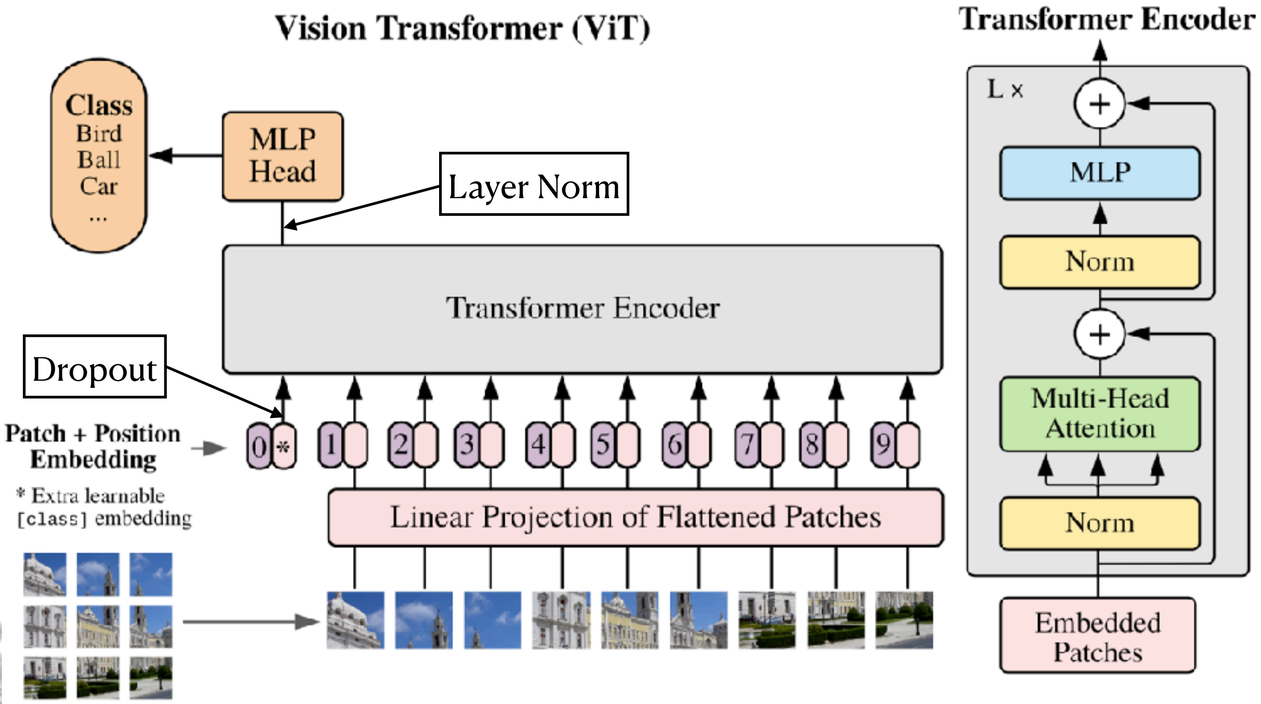


Fig. 3-1 ViT overview

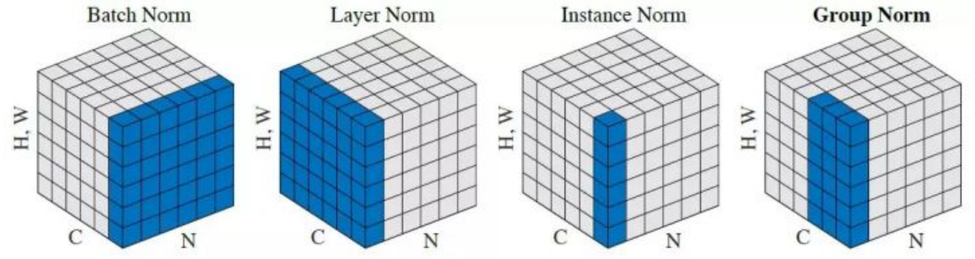


Fig. 3-2 Different norm methods

An image is worth 16*16 Words: transformer for image recognition at scale

Prior knowledge

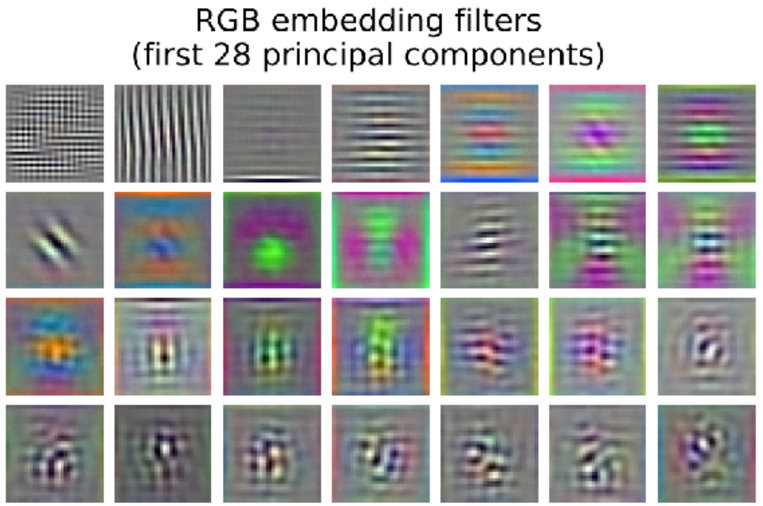


Fig. 3-3 Filters of the init linear embedding of RGB values of ViT-L/32

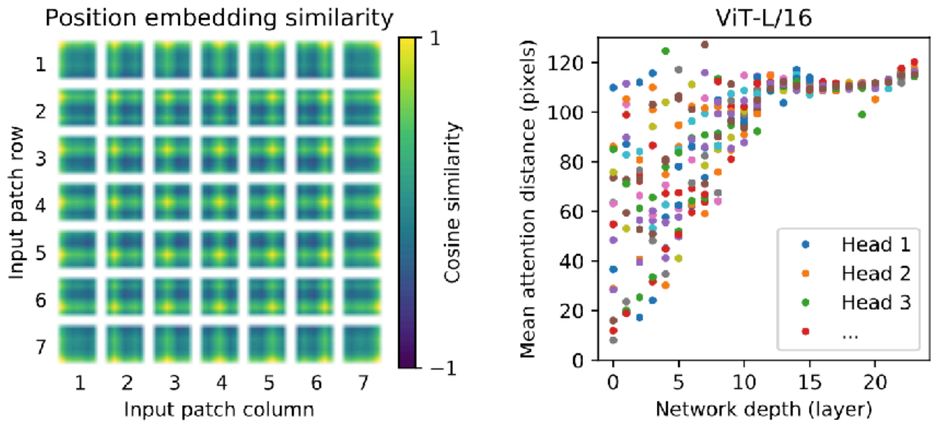


Fig. 3-4 Similarity of position embeddings of ViT-L/32

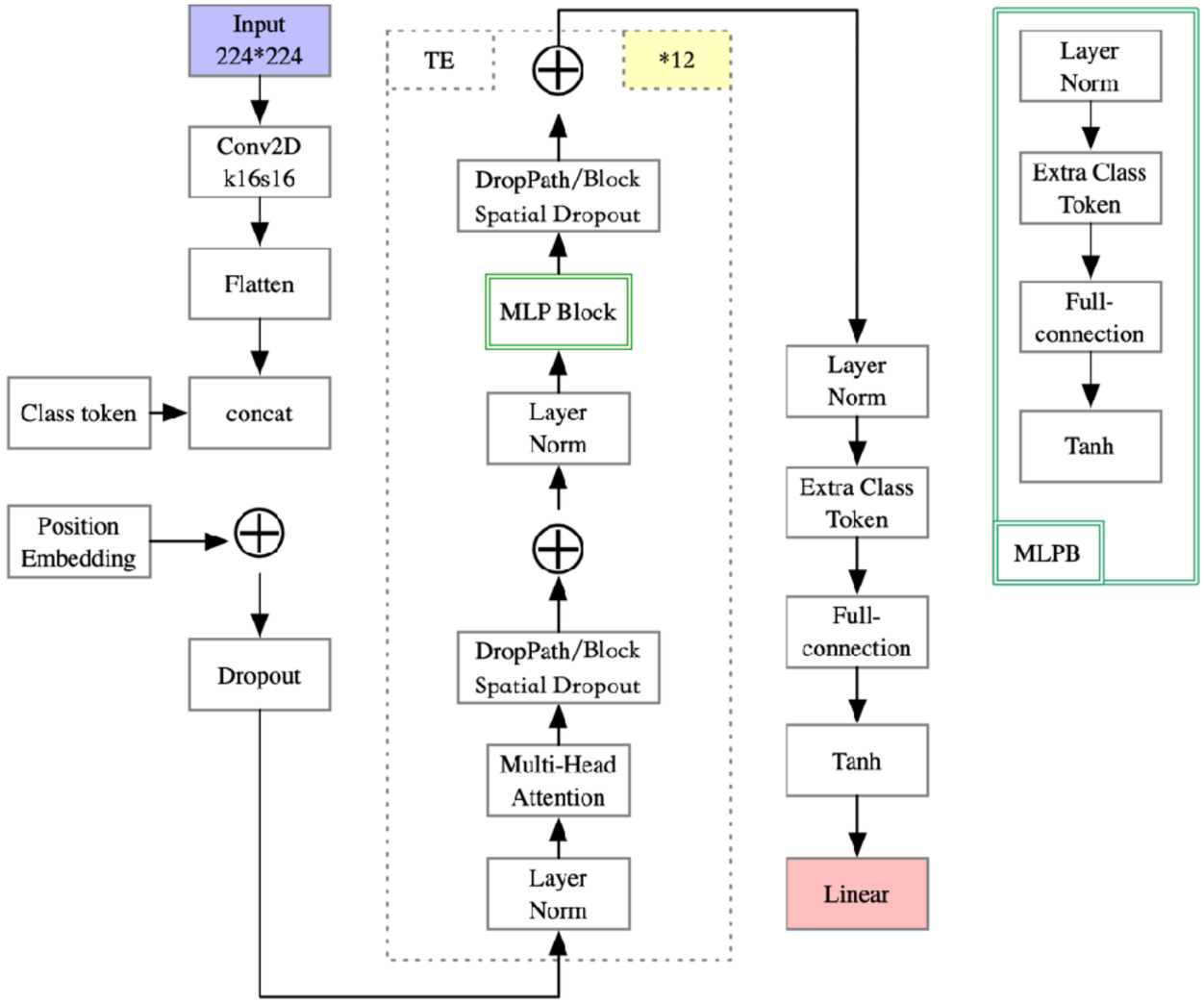


Fig. 3-5 The architecture of ViT

Masked Autoencoder

Architecture

- **Image**-» masked image-» faltten-» encoding-» embedded image + mask token-» decoding-» **Image**

Fig. 3-6 Architecture of MAE

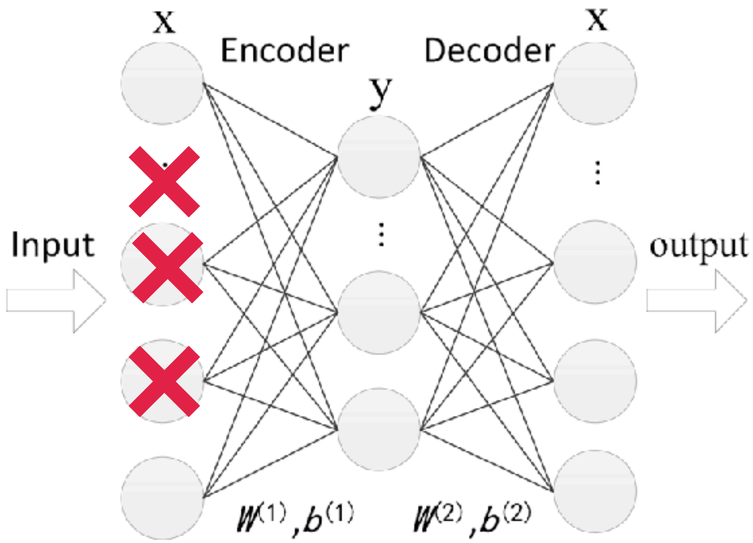


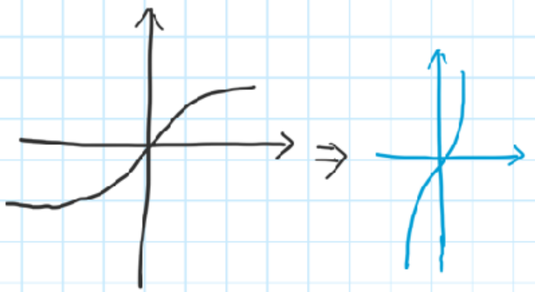
Fig. 3-7 Autoencoder (auto-associator) & Denoise autoencoder

$$y = f(x) = \tanh(W_1 x + b_1)$$

$$x = (\tanh^{-1} y - b_1) \cdot \frac{1}{W_1}$$

$$= act(W_2 y + b_2)$$

$$Loss = MSE + L_p$$



Masked Autoencoders Are Scalable Vision Learners

Masked Autoencoder

- Fine-tune is better.
- Random sampling works better.
- Masked autoencoder has better generalization.

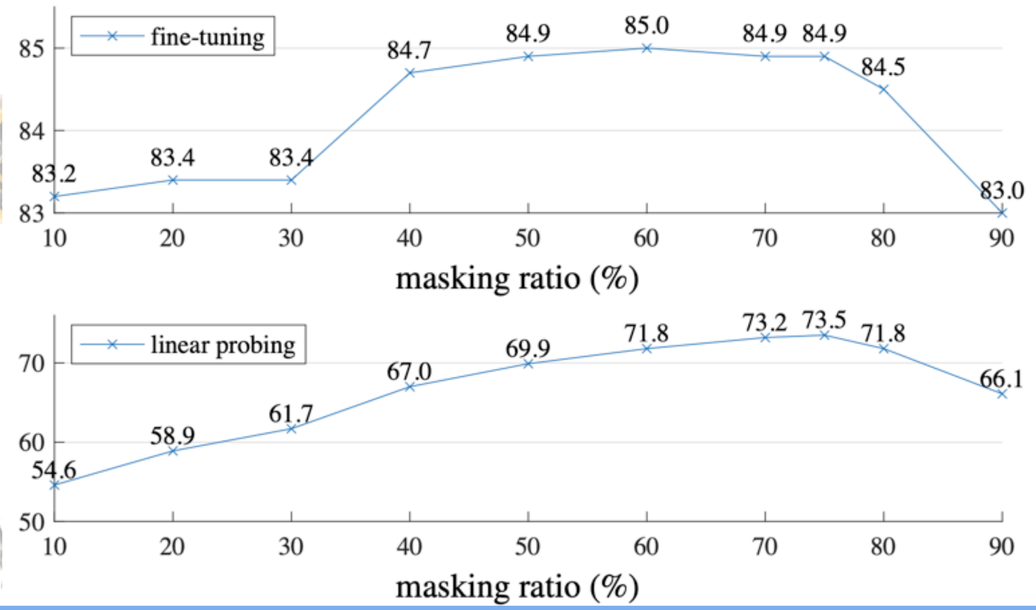


Fig. 3-8 Masking ratio v.s. validation accuracy (Image-1K)

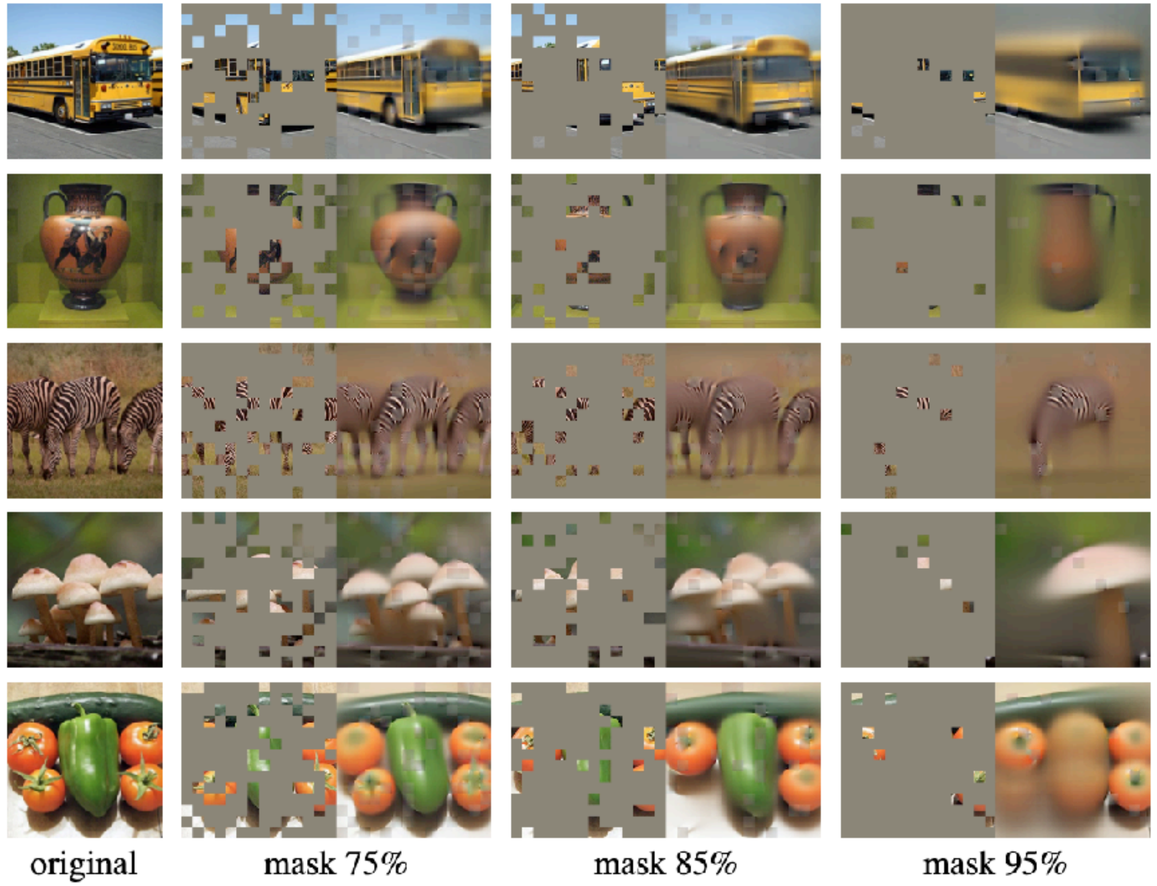


Fig. 3-9 MAE pretrained with 75% masking ratio but applied on inputs with higher masking ratio.

case	ft	lin	FLOPs
encoder w/ [M]	84.2	59.6	3.3×
encoder w/o [M]	**84.9**	**73.5**	**1**×

Tab. 3-1 The comparison between 'with' or 'without' mask tokens

case	ratio	ft	lin
random	75	**84.9**	**73.5**
block	50	83.9	72.3
block	75	82.8	63.9
grid	75	84.0	66.0

Tab. 3-2 Random sampling works best

Masked Autoencoder

Architecture

- **Skipping the mask token in the encoder.**
- **a smaller decoder (1-block), a larger encoder (ViT-H)**
- **Partial Fine-tune**

encoder	dec. depth	ft acc	hours	speedup
ViT-L, w/ [M]	8	84.2	42.4	-
ViT-L	8	84.9	15.4	2.8×
ViT-L	1	84.8	11.6	**3.7**×
ViT-H, w/ [M]	8	-	119.6†	-
ViT-H	8	85.8	34.5	3.5×
ViT-H	1	85.9	29.3	**4.1**×

Tab. 3-3 Wall-clock time of MAE 800 epochs training in 128 TPU-V3 cores with TensorFlow. Decoder width=512. Mask ratio = 75%

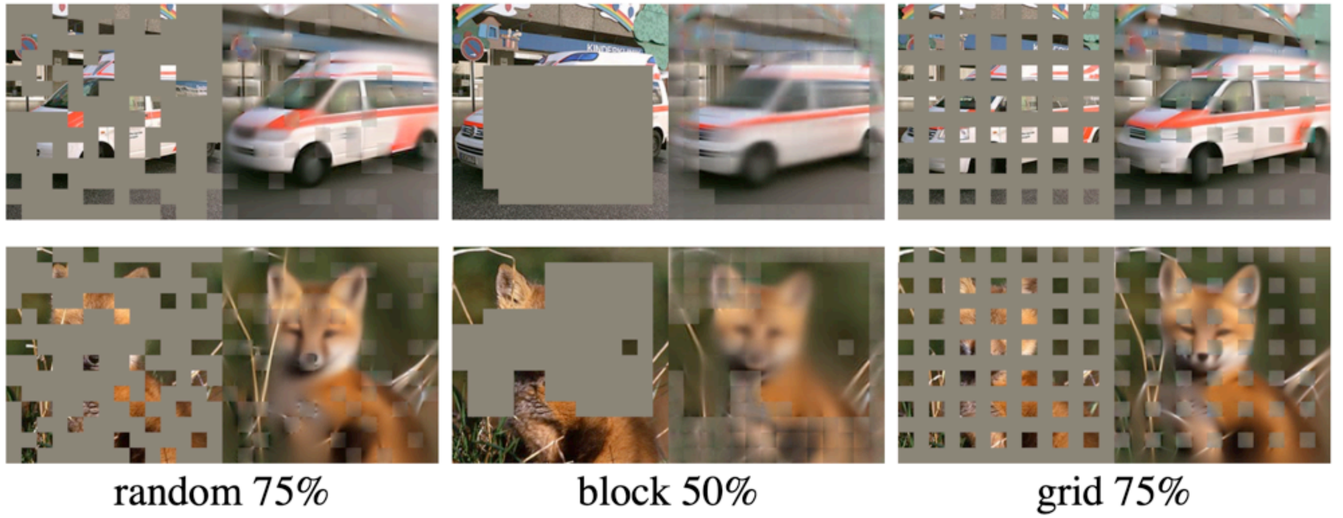


Fig. 3-10 Mask sampling strategies determine the pretext task difficulty, influencing reconstruction quality and representations

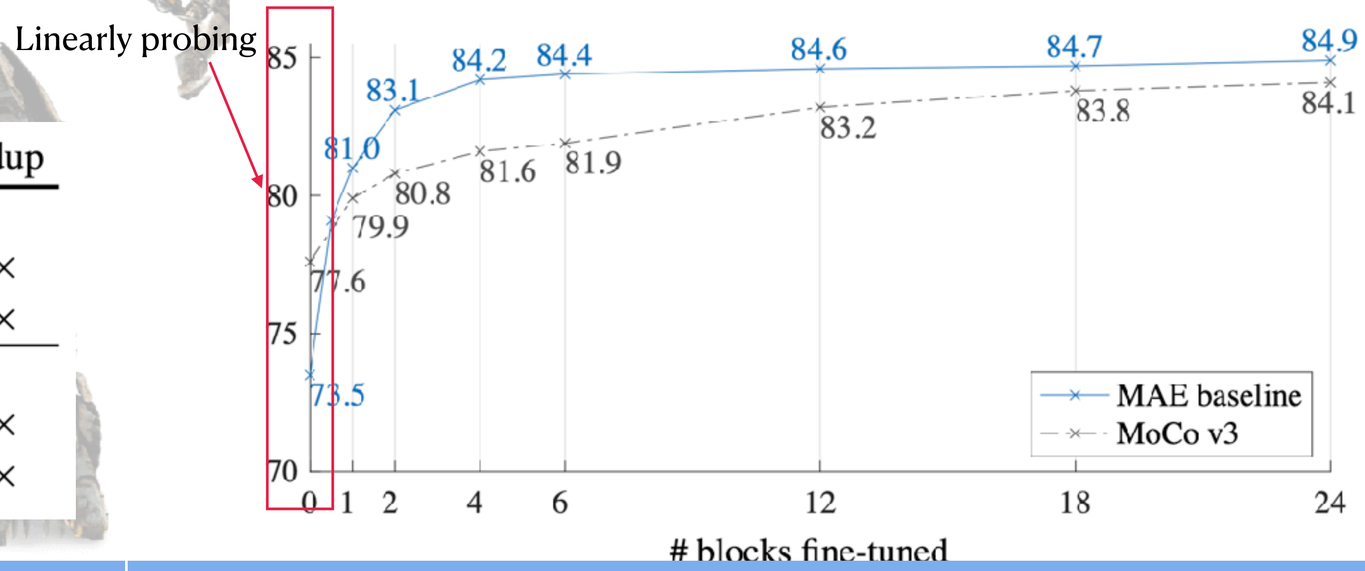


Fig. 3-11 Partial fine tuning, ViT-L using default setting

Conclusion

- **CV is mimicking humans unconsciously.**
- **How to define optics?**

Optics is the branch of physics that studies the behaviour and properties of light, including its interactions with matter and the construction of instruments that use or detect it.[1] Optics usually describes the behaviour of visible, ultraviolet, and infrared light. Because light is an electromagnetic wave, other forms of electromagnetic radiation such as X-rays, microwaves, and radio waves exhibit similar properties.[1]

- **CI—» CV «—ONN**

config	value
optimizer	AdamW [34]
base learning rate	1.5e-4
weight decay	0.05
optimizer momentum	β_1, β_2=0.9, 0.95 [6]
batch size	4096
learning rate schedule	cosine decay [33]
warmup epochs [19]	40
augmentation	RandomResizedCrop

Table 7. **Pre-training setting.**

config	value
optimizer	AdamW
base learning rate	1e-3
weight decay	0.05
optimizer momentum	β_1, β_2=0.9, 0.999
layer-wise lr decay [10, 2]	0.75
batch size	1024
learning rate schedule	cosine decay
warmup epochs	5
training epochs	100 (B), 50 (L/H)
augmentation	RandAug (9, 0.5) [12]
label smoothing [45]	0.1
mixup [58]	0.8
cutmix [57]	1.0
drop path [26]	0.1 (B/L) 0.2 (H)

Table 8. **End-to-end fine-tuning setting.**

config	value
optimizer	LARS [55]
base learning rate	0.1
weight decay	0
optimizer momentum	0.9
batch size	16384
learning rate schedule	cosine decay
warmup epochs	10
training epochs	90
augmentation	RandomResizedCrop

Table 9. **Linear probing setting.** We use LARS with a large batch for faster training; SGD works similarly with a 4096 batch size.

Outline

- **Analogy of ONN**
- ONN implementation
- A glimpse
- A optics inspired design

Preliminary

Gerchberg-Saxton algorithm

- **To find a phase that conforms to the measurements of I & O.**

- $$\boldsymbol{U}(x,y;d) = \mathcal{F}^{-1}\left(\mathcal{F}(\boldsymbol{U}(x,y;0)) \odot \boldsymbol{G}(f_x,f_y)\right) = \mathrm{H}\left(\boldsymbol{U}(x,y;0)\right) \quad (1)$$

$$\boldsymbol{G}(f_x,f_y) = \exp\left[i\boldsymbol{k}d\sqrt{1-(\lambda f_x)^2-(\lambda f_y)^2}\right] \quad (2)$$

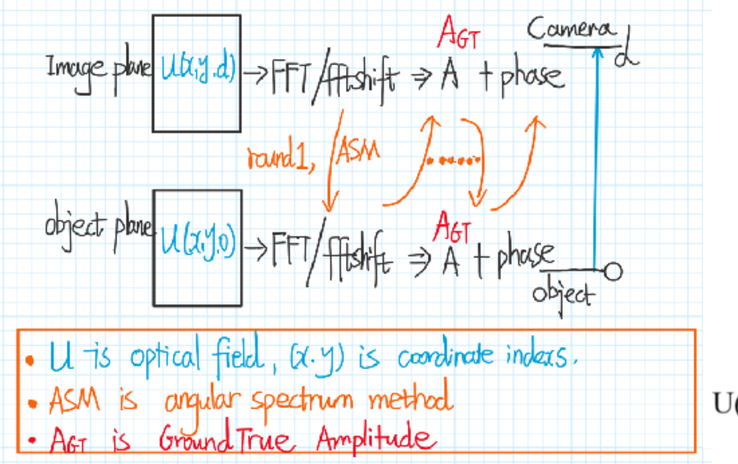
$$\boldsymbol{Y}(x,y;d) = |\boldsymbol{U}(x,y;d)|^2 = |\mathrm{H}\left(\boldsymbol{U}(x,y;0)\right)|^2 \quad (3)$$

$$\hat{\boldsymbol{U}}(x,y,0) = \underset{\boldsymbol{U}(x,y;0)}{\arg\min} \|\boldsymbol{Y}(x,y;d) - |\mathrm{H}(\boldsymbol{U}(x,y;0))|^2\|_2^2 \quad (4)$$

$$\boldsymbol{I}^{(k)}(x,y;d) = \sqrt{\boldsymbol{Y}(x,y;d)}\cdot\exp[\boldsymbol{i}\cdot\mathrm{angle}(\boldsymbol{U}^{(k)}(x,y;d))]$$

$$\boldsymbol{O}^{(k)}(x,y;0) = \mathrm{H}^{-1}(\boldsymbol{I}^{(k)}(x,y;d))$$

$$\boldsymbol{U}^{(k+1)}(x,y;0) = \boldsymbol{P}(x,y;0)\cdot\exp[\boldsymbol{i}\cdot\mathrm{angle}(\boldsymbol{O}^{(k)}(x,y;0))]$$



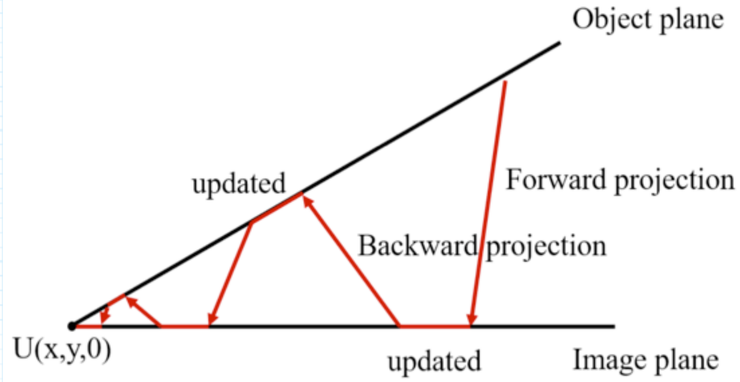


Fig. 1 The process of Gerchberg-Saxton algorithm

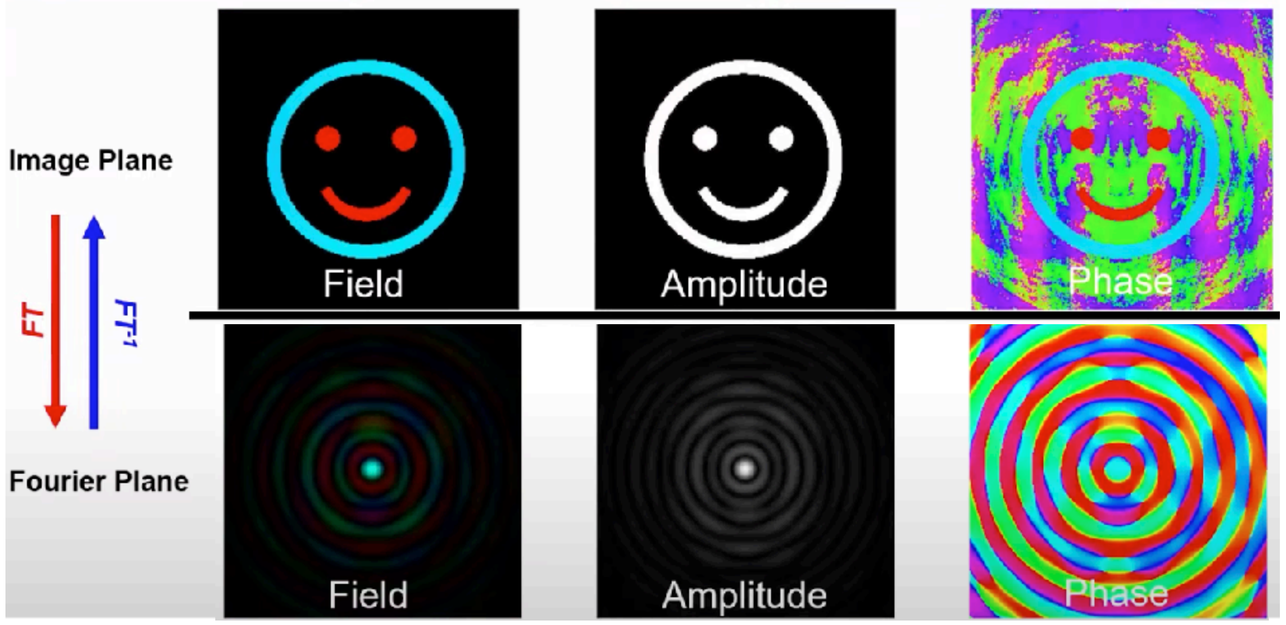


Fig. 2 Illustration of GS algorithm

Preliminary

Angular spectrum method

- **Angular spectrum method:**
  1. Decompose U(x, y, 0) into 2d plane waves;
  2. Propagate each plane wave separately to the plane d;
  3. Add the propagated plane waves together (U(x, y, d))
- Where the U is optical field, k is wave vector, f is frequency, and F is fourier transformation

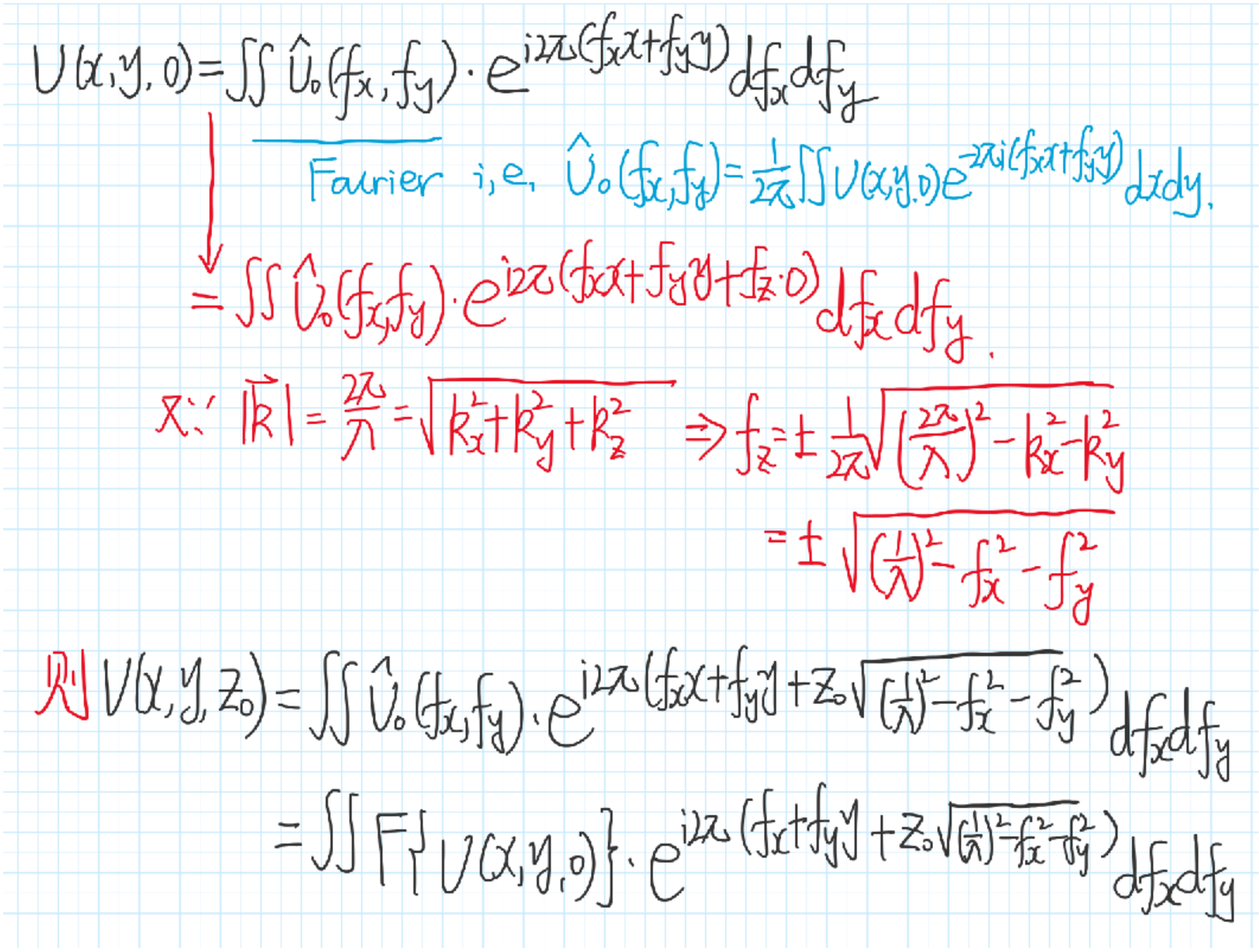


Fig. 3 The process of Gerchberg-Saxton algorithm

Preliminary

compressed sensing

- Conditions: **Sparsity** and **incoherence**

$$min_{\mathrm{X}}\|\nabla \mathrm{X} \bullet d\|_1 + \frac{\lambda}{2}\|Y - \Phi \mathrm{X}\|_2^2$$

- where the d = (dh,dv) define the horizontal and vertical estimates of d
- ΔX = grad X, X is Object & Y is measurement

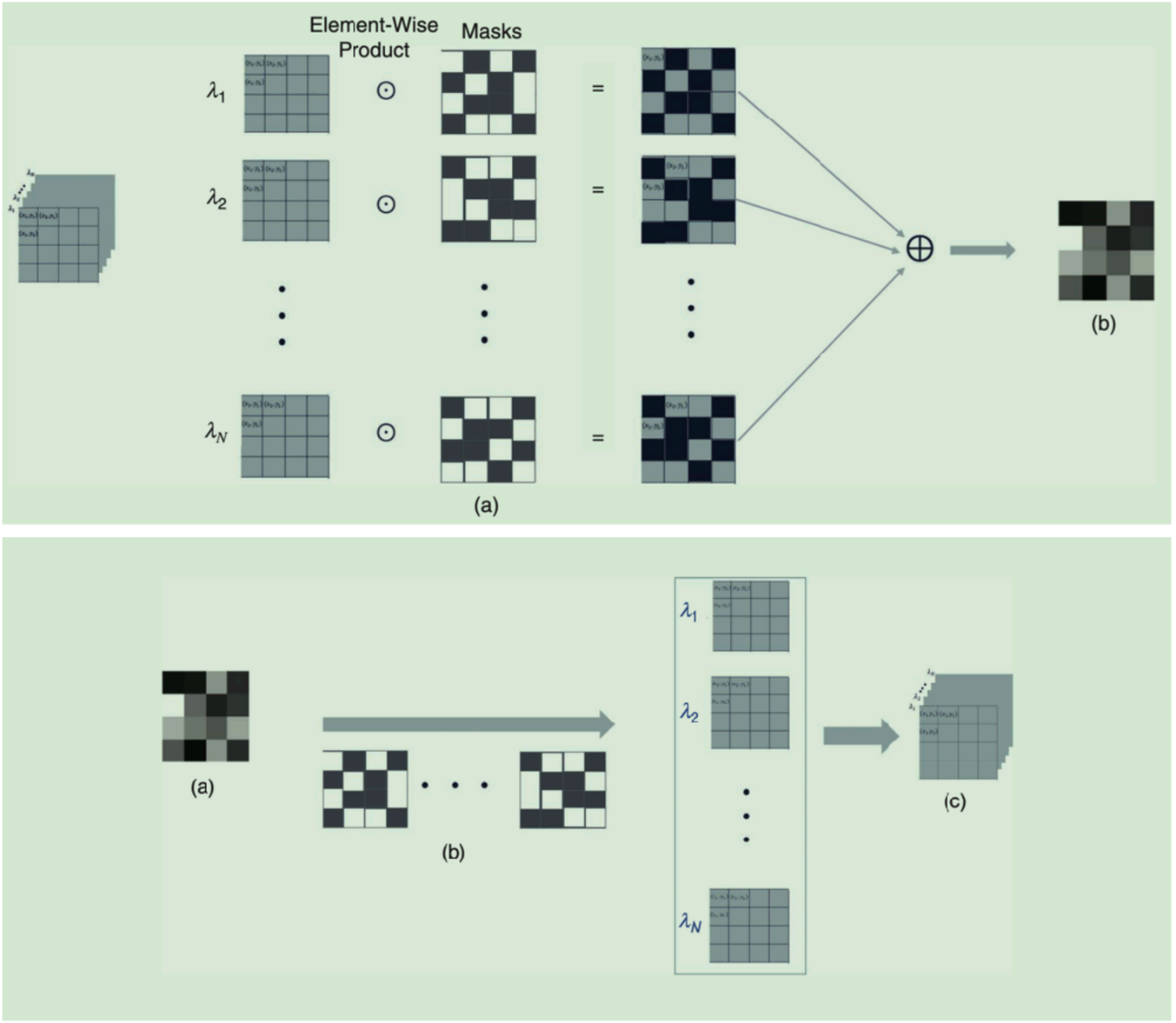


Fig. 4 (x,y,lambda) compressed sensing

PR in Lensless microscopy imaging

Current challenges

1. Limited dataset
2. Difficult to Supervise
3. Additional hardware

Schematic

- a is setup of lensless microscopy imaging
- b is structure diagram of the proposed NN
- c is comple-valued U-net (2D)

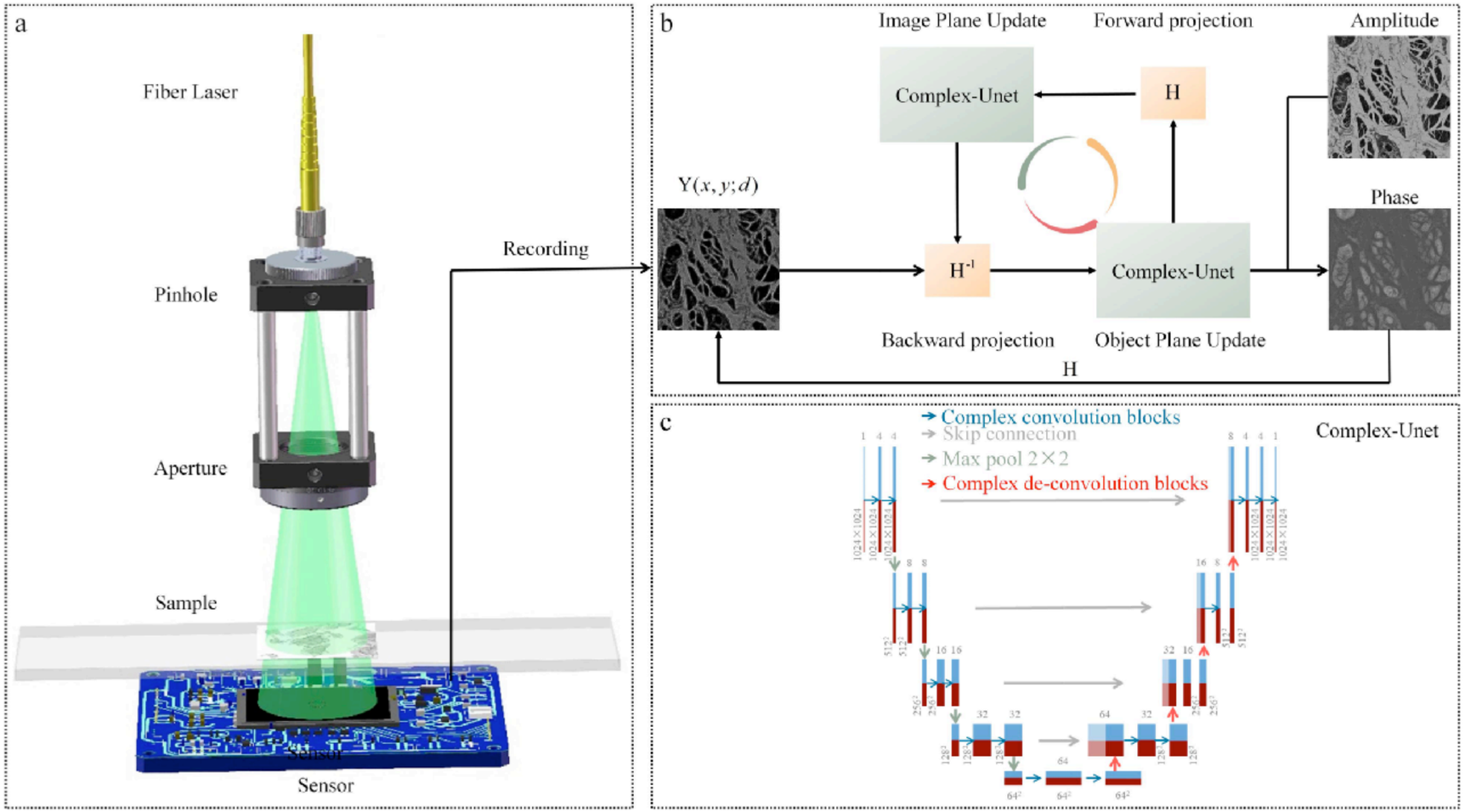


Fig. 5 Schematic diagram of the imaging process

PR in Lensless microscopy imaging

- Backward—>forward—>Backward
- Constraint : ASM and I from CMOS
- Complex-valued UNet:

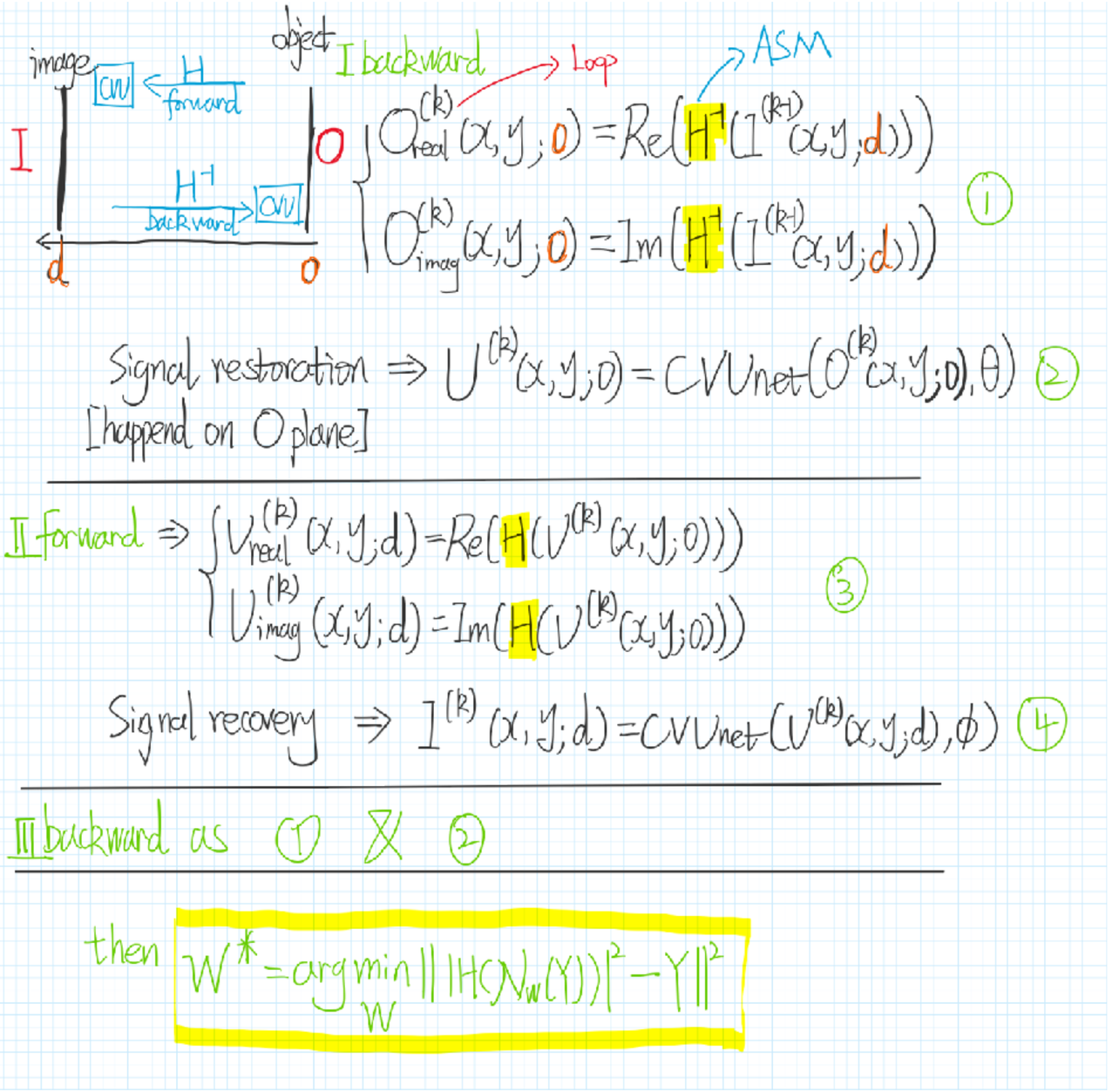
Convolution

$$\begin{aligned}\boldsymbol{F} * \boldsymbol{h} &= (\boldsymbol{F}_r + \boldsymbol{i}\boldsymbol{F}_i) * (\boldsymbol{h}_r + \boldsymbol{i}\boldsymbol{h}_i) \\ &= (\boldsymbol{F}_r * \boldsymbol{h}_r - \boldsymbol{F}_i * \boldsymbol{h}_i) + \boldsymbol{i}(\boldsymbol{F}_r * \boldsymbol{h}_i + \boldsymbol{F}_i * \boldsymbol{h}_r)\end{aligned}$$

LeakReLU

$$\mathbb{C}LeReLU(\boldsymbol{z}) = LeReLU(\boldsymbol{z}_r) + \boldsymbol{i}LeReLU(\boldsymbol{z}_i)$$

Why LeakReLU?



PR in Lensless microscopy imaging

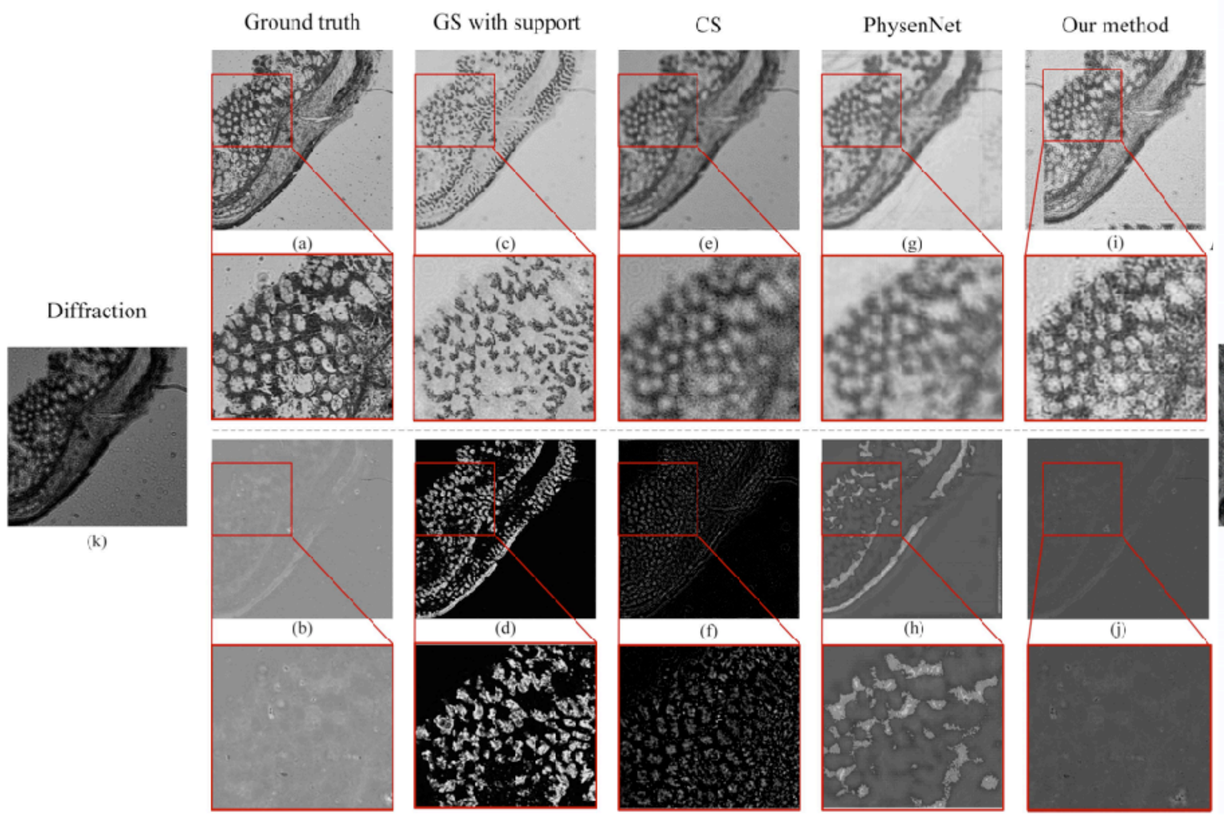


Fig. 6 Visual comparisons on the H&E stained pathological slides of rat intestine

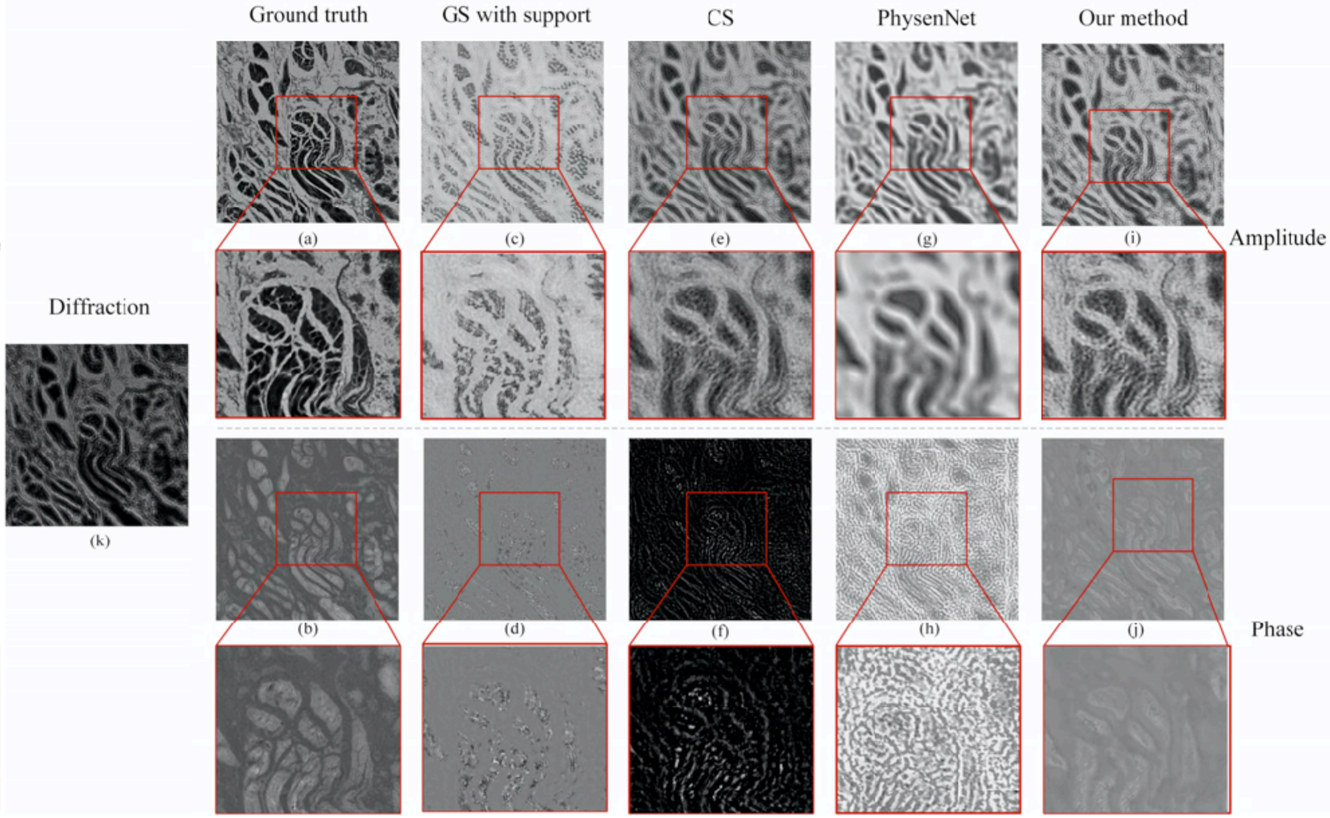
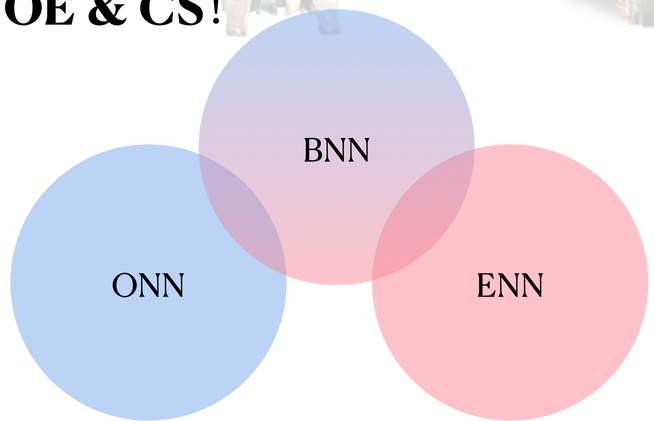


Fig. 6 Visual comparisons on the H&E stained pathological slides of human esophagus cancer cell

Conclusion

- OC: or photonic computing, using photons (from correlation source) for computation
- CI: process of indirectly imaging from measurements using algos that rely on computing.
- ONN: a physical implementation of an artificial neural network with optical components.
- BNNs function on an electrochemical basis. ONNs use electromagnetic waves. ENNs are based on simulation.
- We can interchange techiques in OE & CS!



Outline

- **Analogy of ONN (E2O | imaging)**
- ONN implementation (E2OE | reasoning)
- A glimpse (B2E | reasoning)
- A optics inspired design (O2E | imaging)