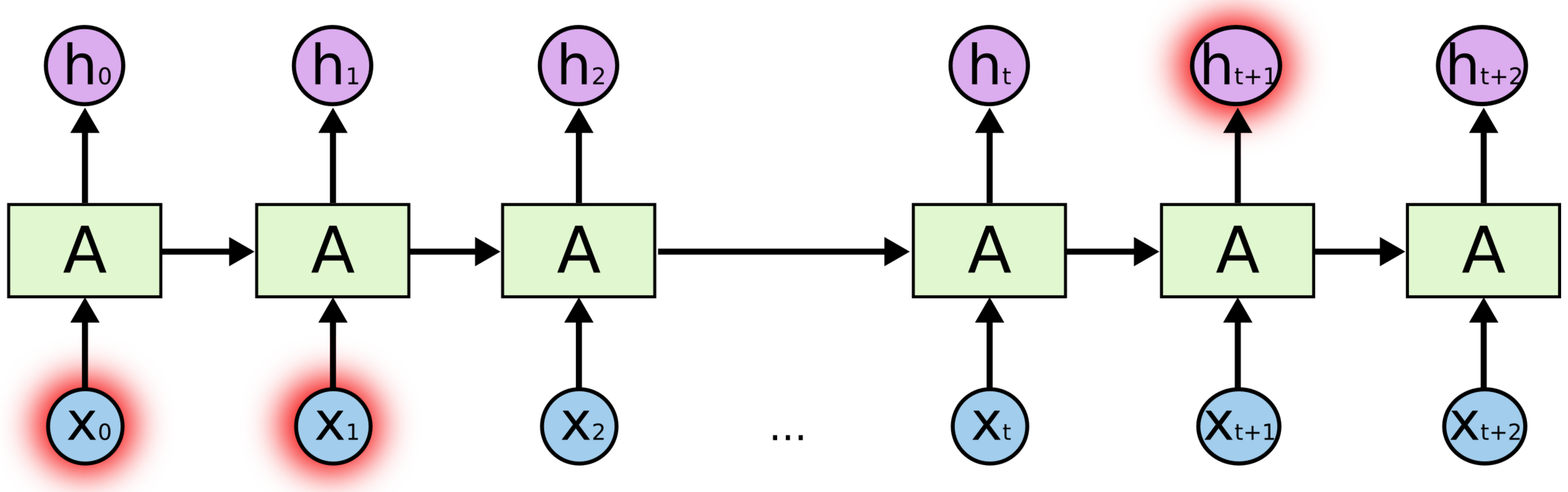
Tunable Efficient Unitary Neural Networks (EUNN) and their application to RNNs

Li Jing*, Yichen Shen*, Tena Dubček, John Peurifoy, Scott Skirlo,
Yann LeCun, Max Tegmark, Marin Soljačić

** equal contribution*

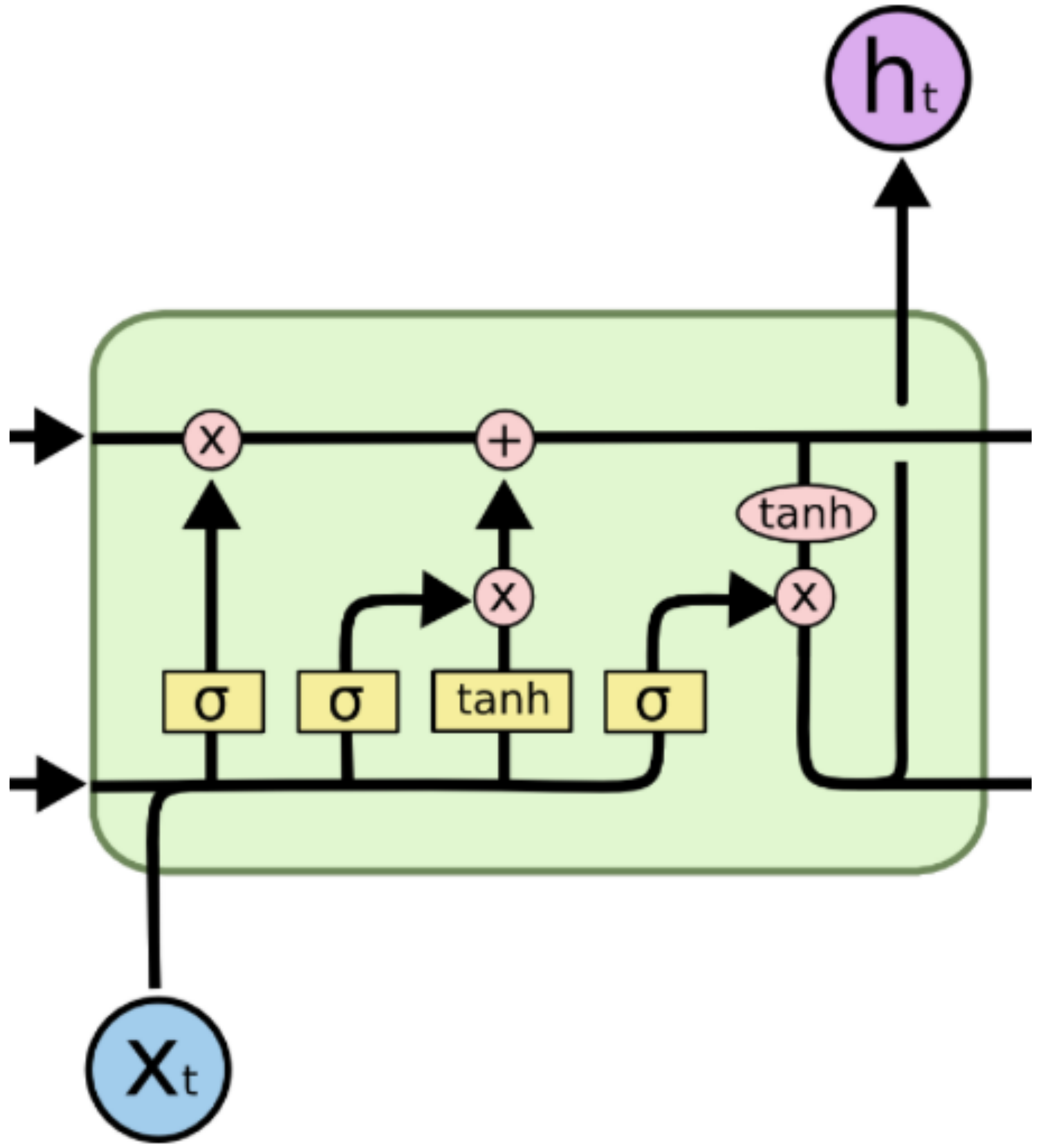
Massachusetts Institute of Technology

Gradient Vanishing/Explosion Problem



- During backpropagation through time, hidden to hidden Jacobin matrix is multiplied multiple times.
- Gradient vanishing/explosion makes RNN hard to train

Conventional Solution: LSTM



- Practically, gradient clipping is required
- slow to learn long term dependency

New Solution: Unitary/Orthogonal RNN

Unitary/Orthogonal matrices keep the norm of vectors: $\|\boldsymbol{U}\boldsymbol{x}\| = \|\boldsymbol{x}\|$

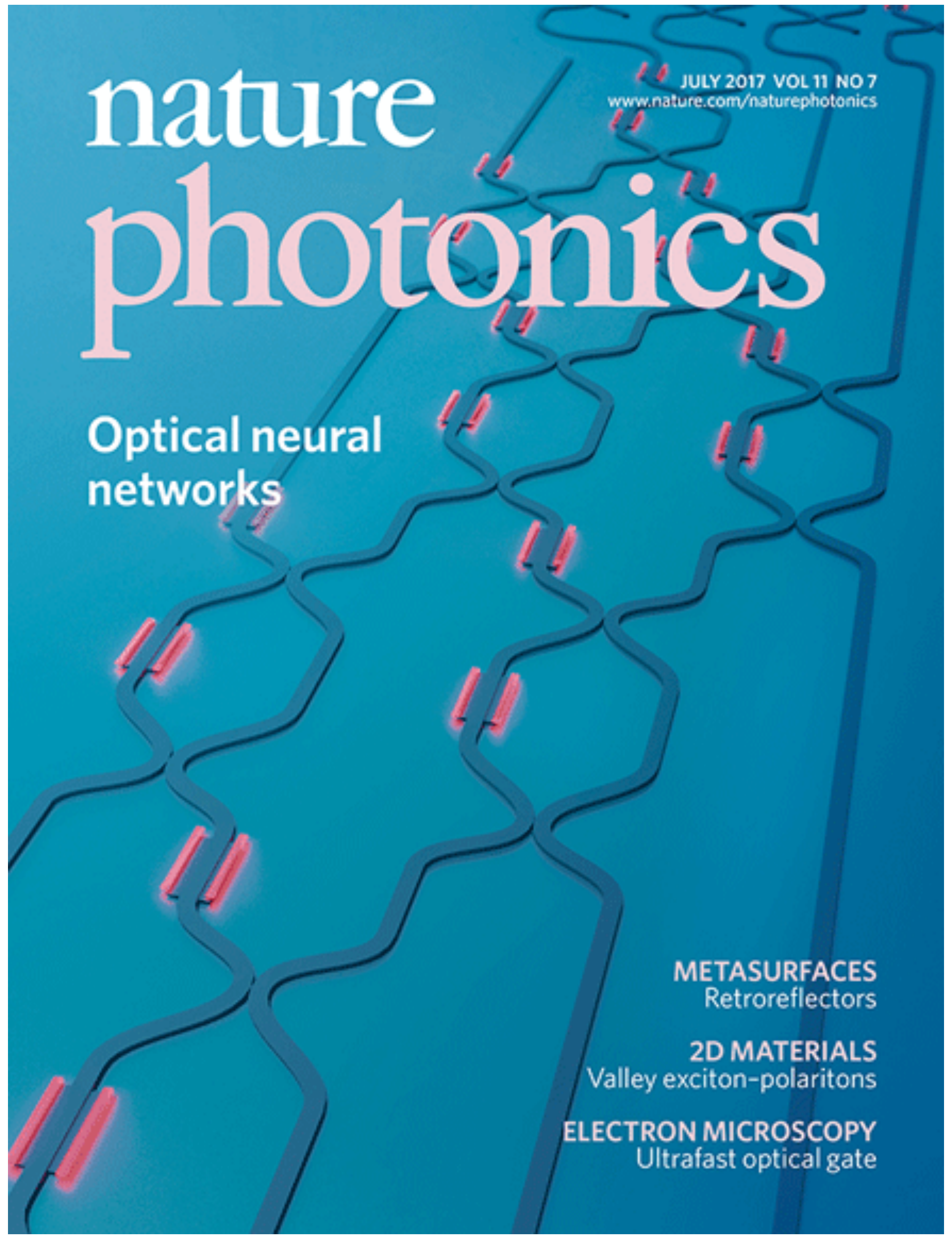
By enforcing hidden to hidden transition matrix to be unitary/orthogonal, no matter how many time steps are propagated, the norm of the gradient will stay the same

$$\left\| \prod_{k=t}^{T-1} \frac{\partial \mathbf{h}^{(k+1)}}{\partial \mathbf{h}^{(k)}} \right\| \sim 1$$

Related Works

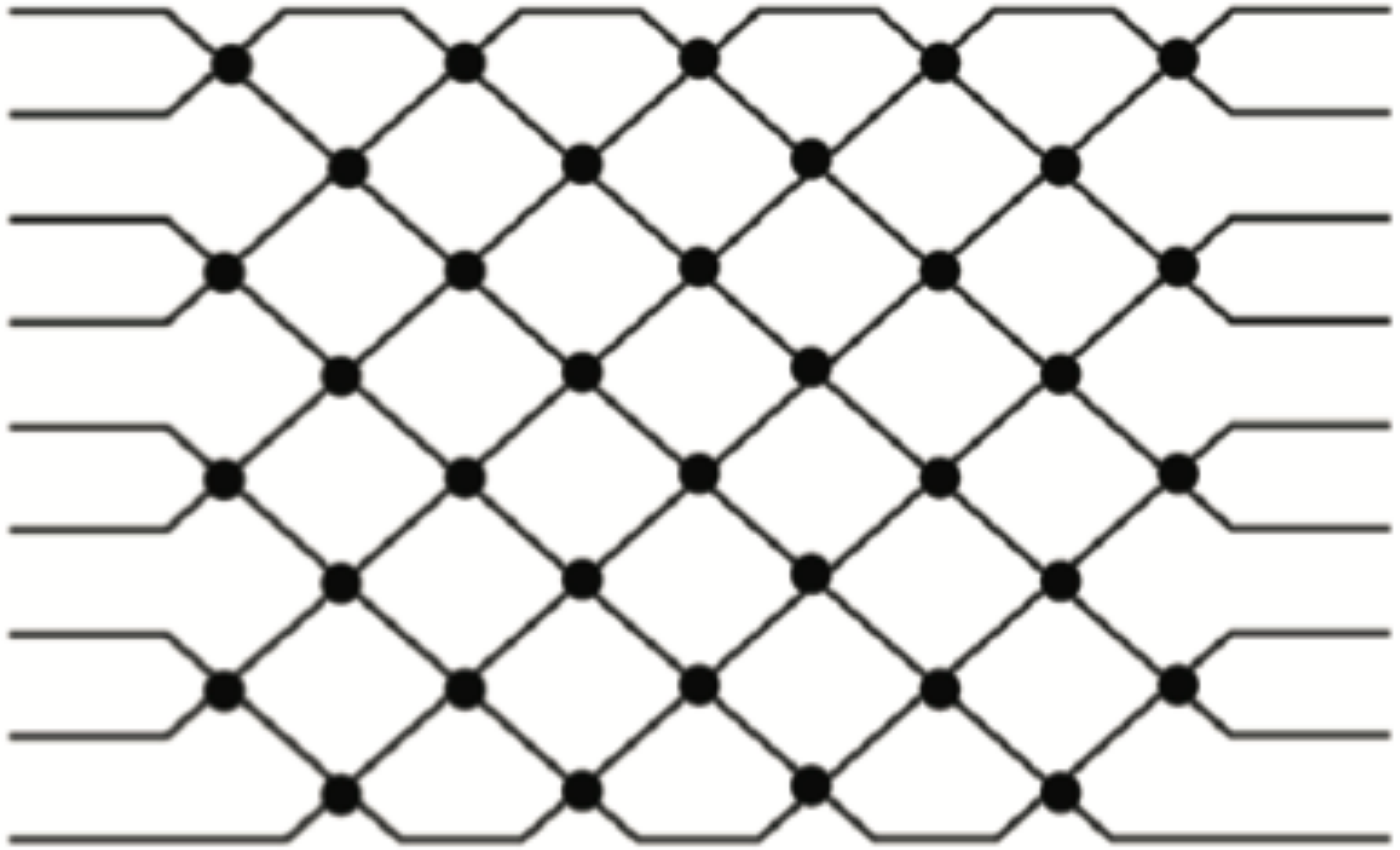
- Restricted-capacity Unitary Matrix Parametrization (Arjovsky, ICML 2016)
- Full-capacity Unitary Matrix by projection (Wisdom, NIPS 2016)
- Orthogonal Matrix Parametrization by reflection (Mhammedi, ICML 2017)
- Orthogonal Matrix by regularization (Vorontsov, ICML 2017)

Efficient Unitary Matrix Parametrization

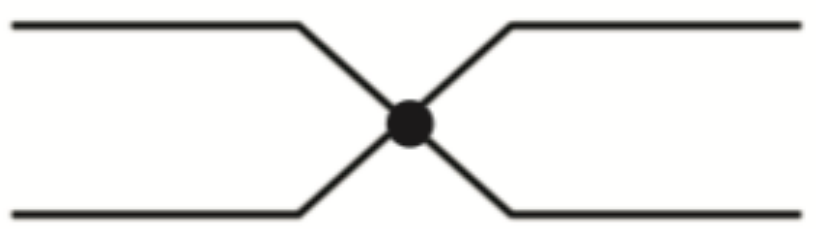


(Y. Shen, et al, Nature Photonics 2017)

Physical system inspired:



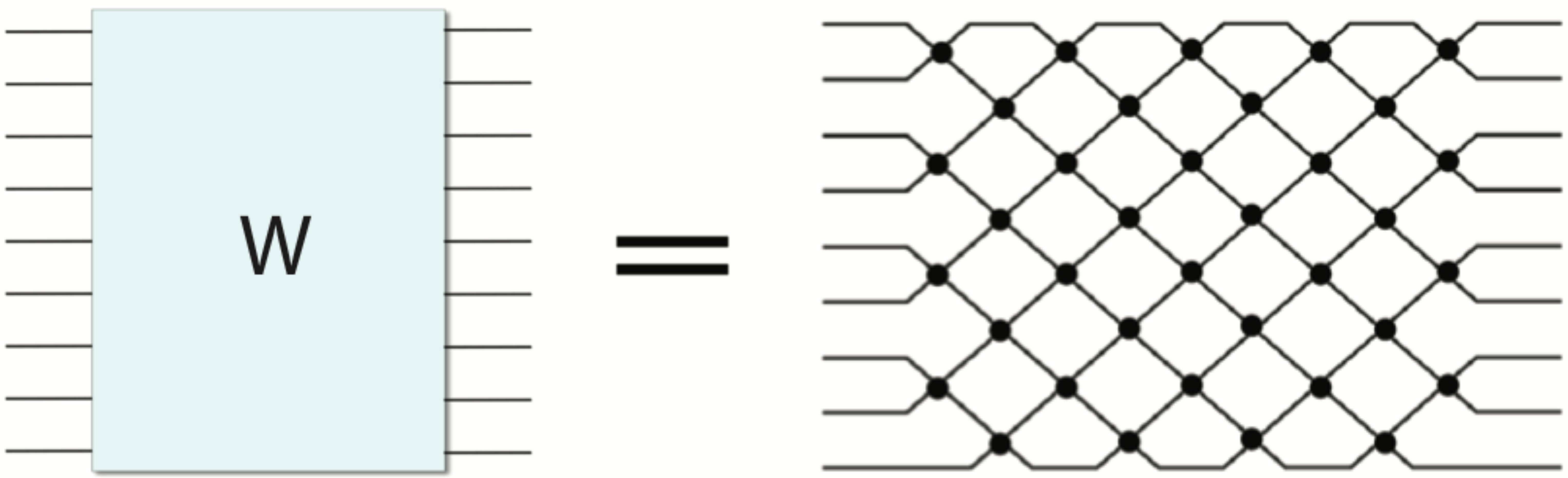
Efficient Unitary Matrix Parametrization



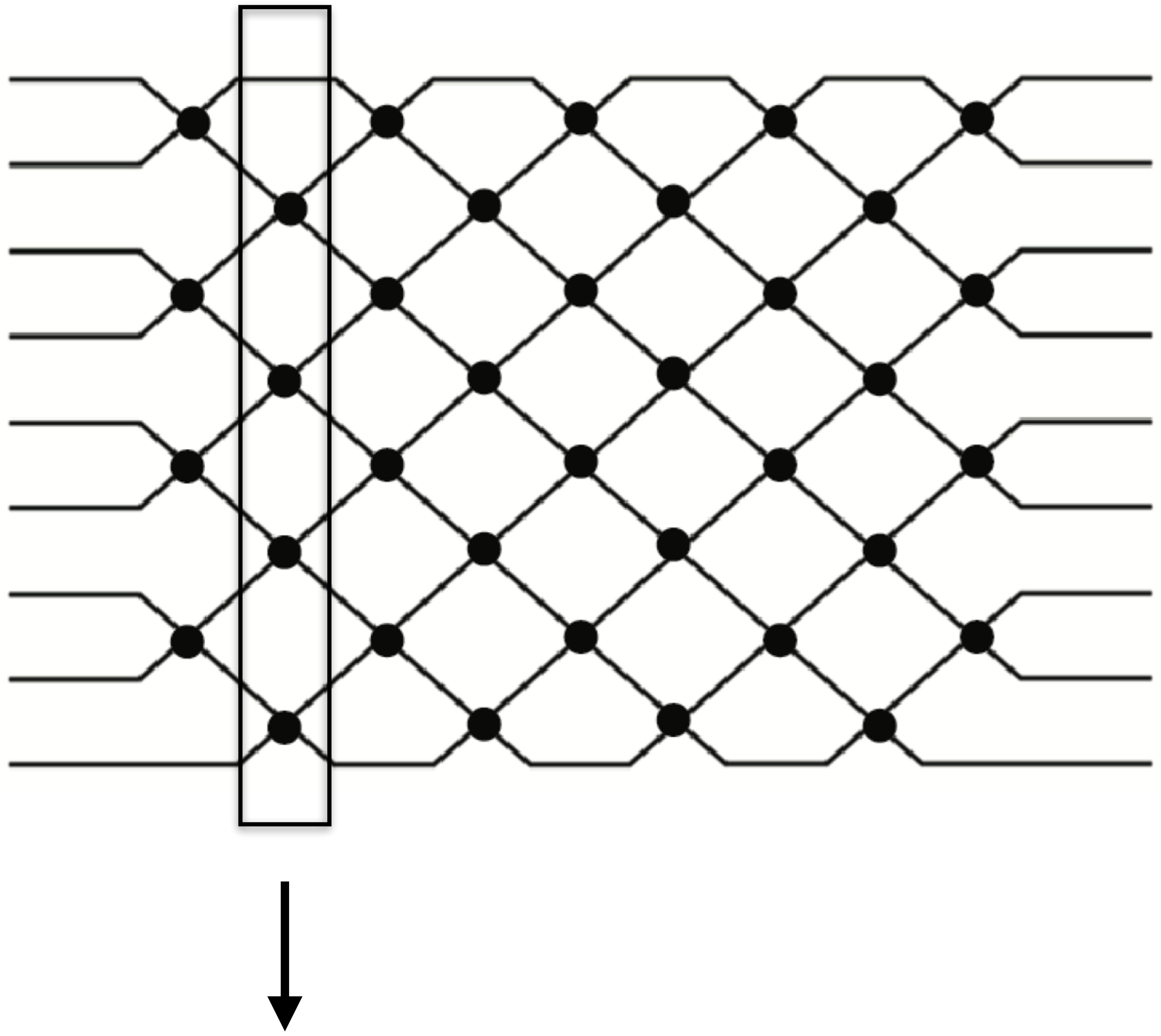
$$= \begin{pmatrix} e^{i\phi}\cos\theta & -e^{i\phi}\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$$

2-by-2 element

- SU(2) element: rotation matrix + relevant phase
- strict, complete, unique
- Full unitary space



Efficient Unitary Matrix Implementation



sparse block diagonal matrix

element-wise functions

Algorithm 1 Efficient implementation for $\mathbf{F}$ with parameter θ_i and ϕ_i.

Input: input $\mathbf{x}$, size N; parameters θ and ϕ, size $N/2$; constant permuatation index list $\mathbf{ind_1}$ and $\mathbf{ind_2}$.
Output: output $\mathbf{y}$, size N.
$\mathbf{v_1} \leftarrow$ concatenate($\cos\theta$, $\cos\theta * \exp(i\phi)$)
$\mathbf{v_2} \leftarrow$ concatenate($\sin\theta$, - $\sin\theta * \exp(i\phi)$)
$\mathbf{v_1} \leftarrow \text{permute}(\mathbf{v_1}, \mathbf{ind_1})$
$\mathbf{v_2} \leftarrow \text{permute}(\mathbf{v_2}, \mathbf{ind_1})$
$\mathbf{y} \leftarrow \mathbf{v_1} * \mathbf{x} + \mathbf{v_2} * \text{permute}(\mathbf{x}, \mathbf{ind_2})$

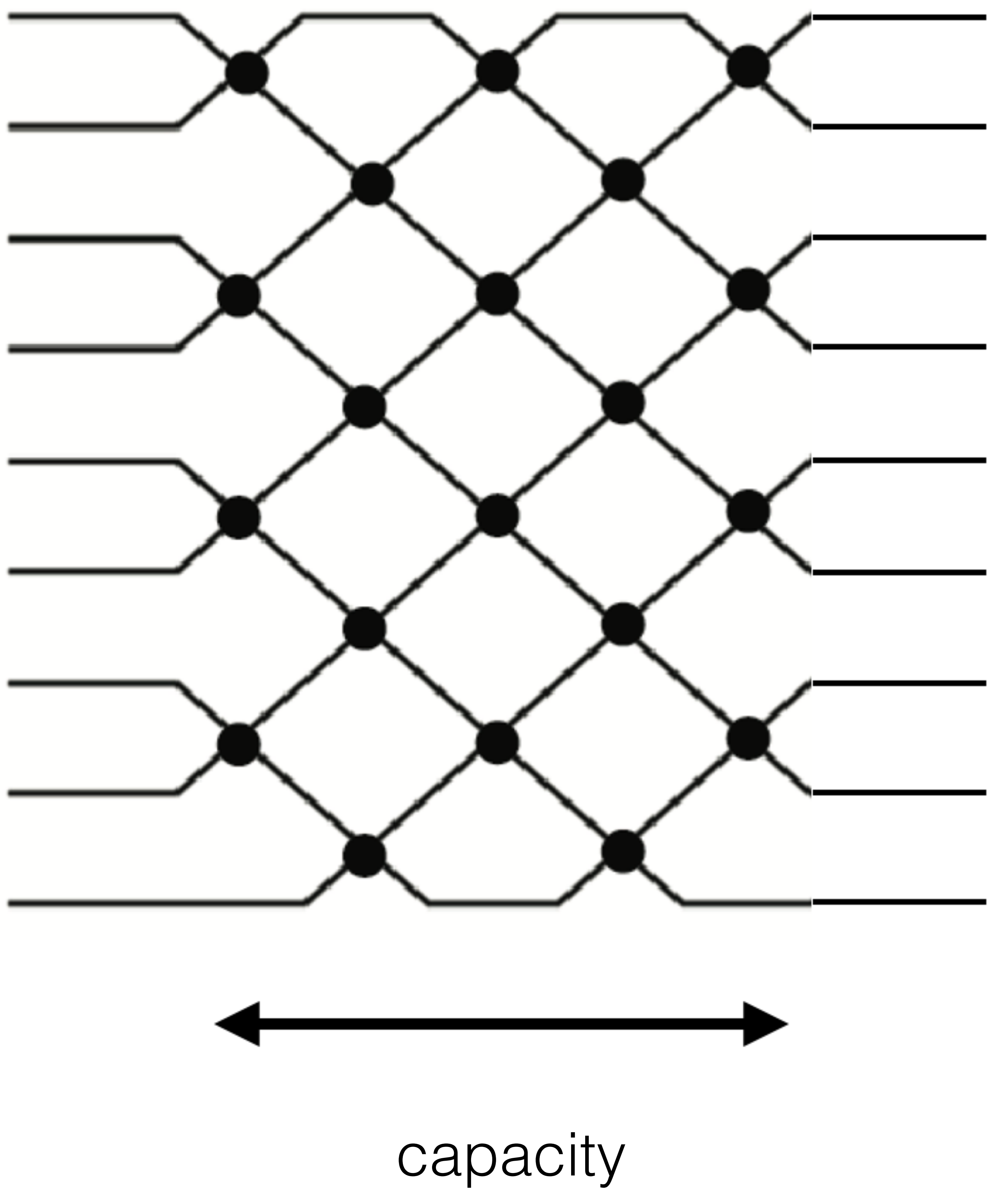
- Parallel, Sparse
- O(1) per parameter
- No need to customize gradient

Tensorflow: https://github.com/jingli9111/EUNN-tensorflow

PyTorch: https://github.com/jingli9111/URNN-PyTorc

Theano: https://github.com/iguanaus/EUNN-theano

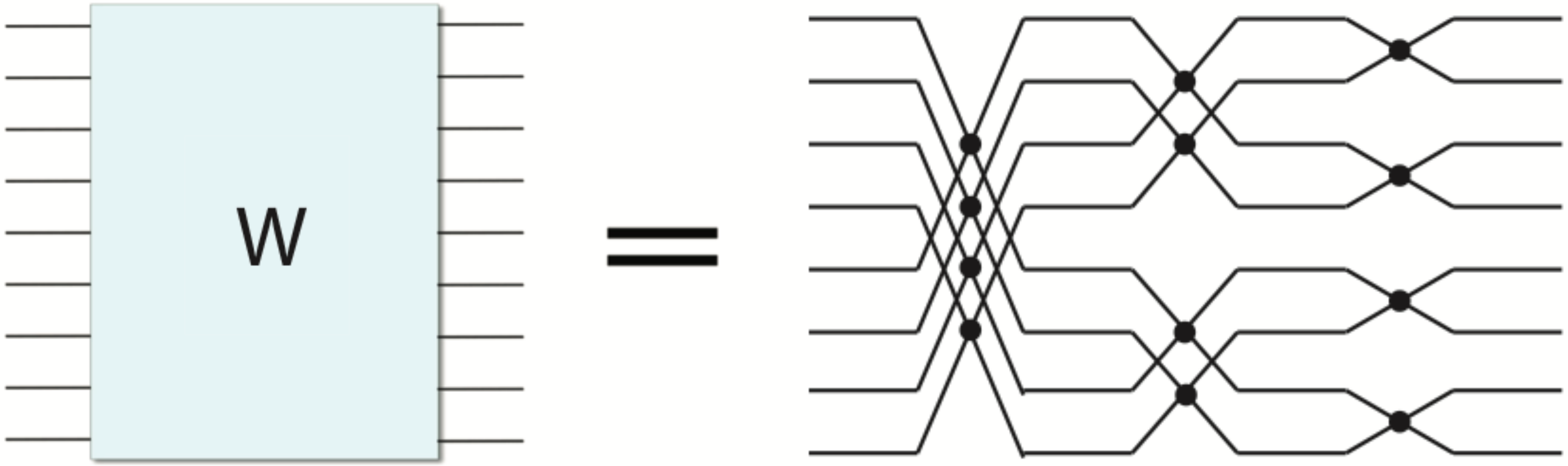
Tunable Parametrization



- reduce capacity allows larger hidden state
- reduce capacity increases training speed

FFT-style Parametrization

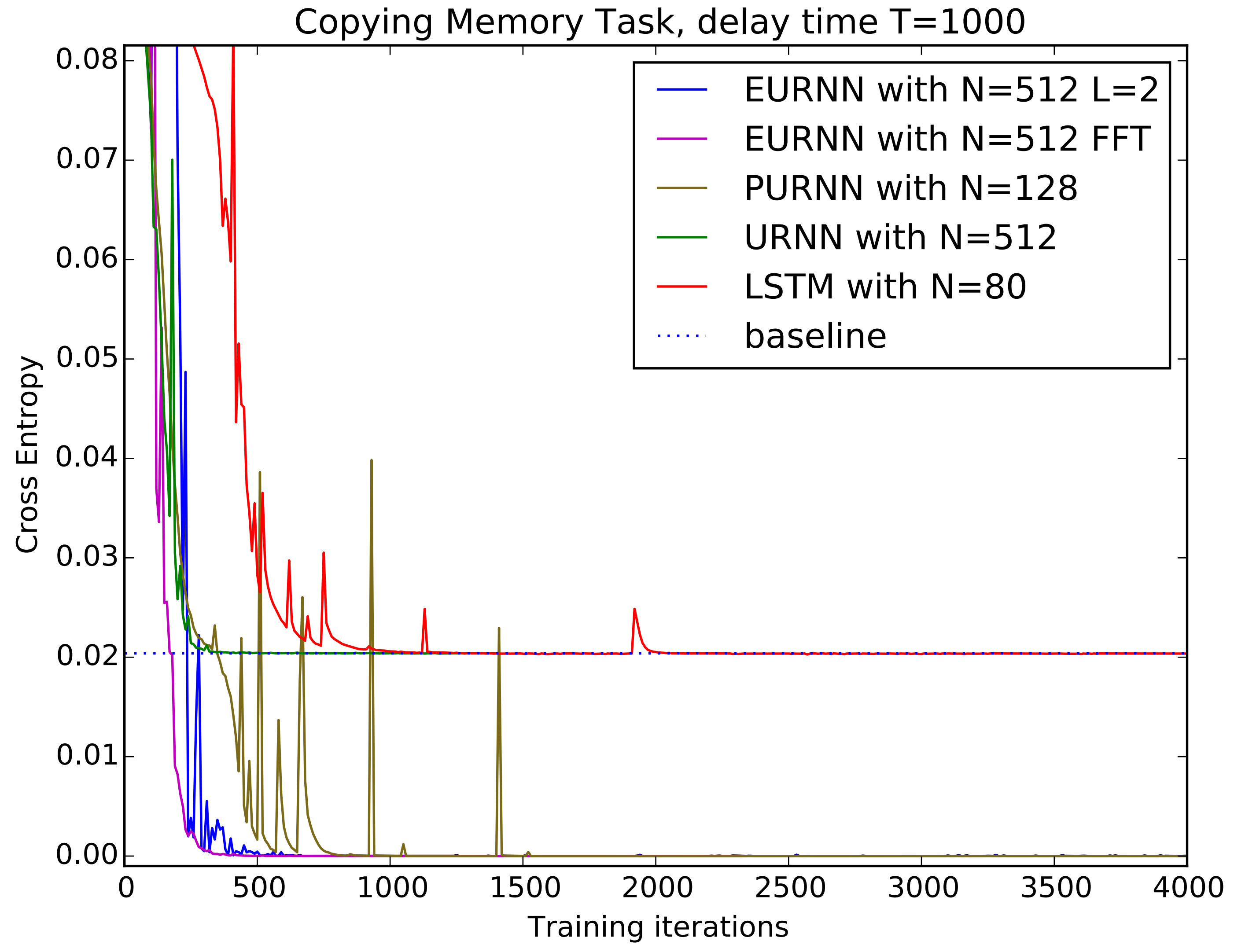
$$= \begin{pmatrix} e^{i\phi}\cos\theta & -e^{i\phi}\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$$



- nlog(n) parameters
- minimum number of parameters for symmetric unitary space
- almost same performance as Tunable style with large capacity

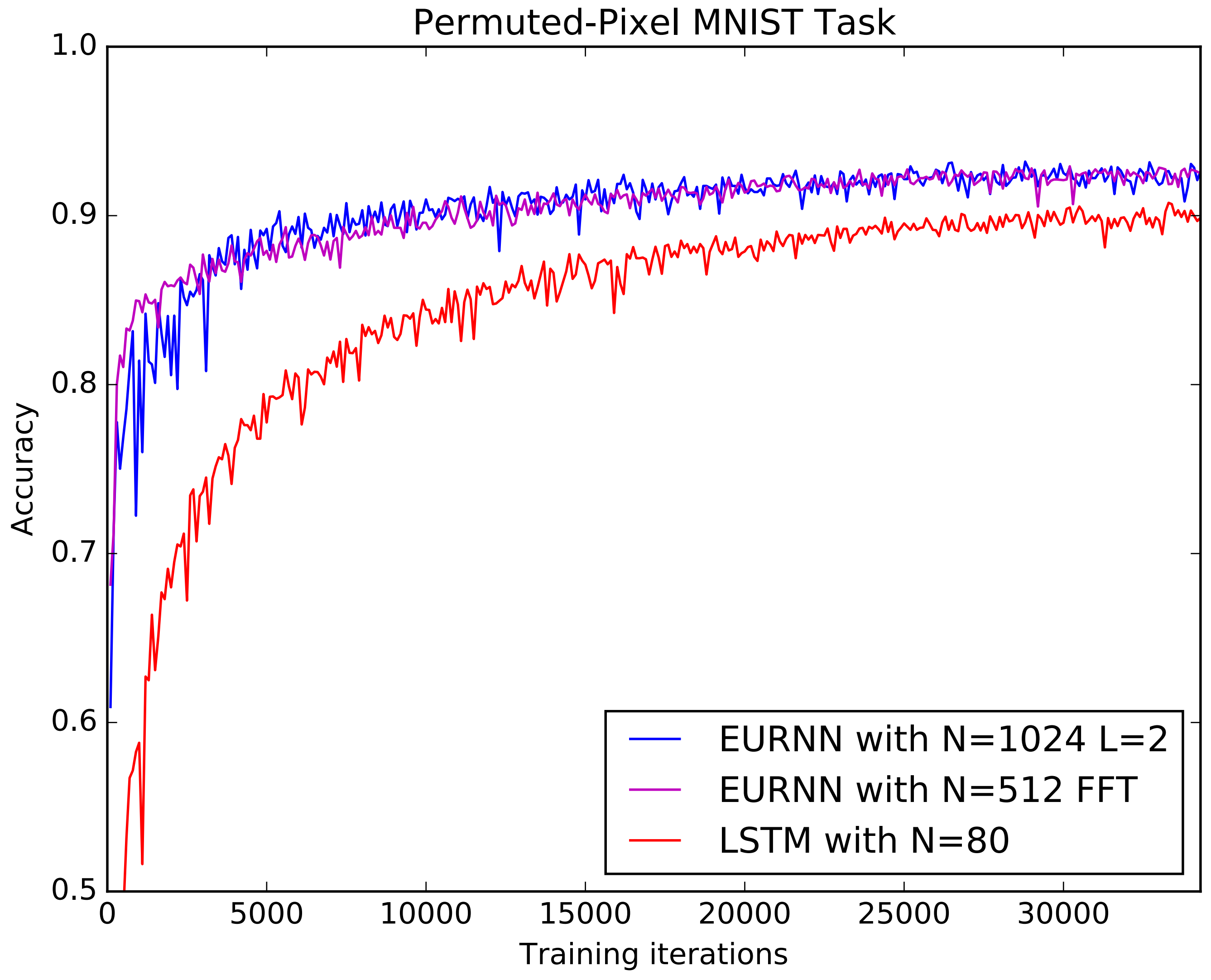
Experiment results

- Copying Memory



EURNN outperforms all other models for long delay time case.

- Pixel Permuted MNIST



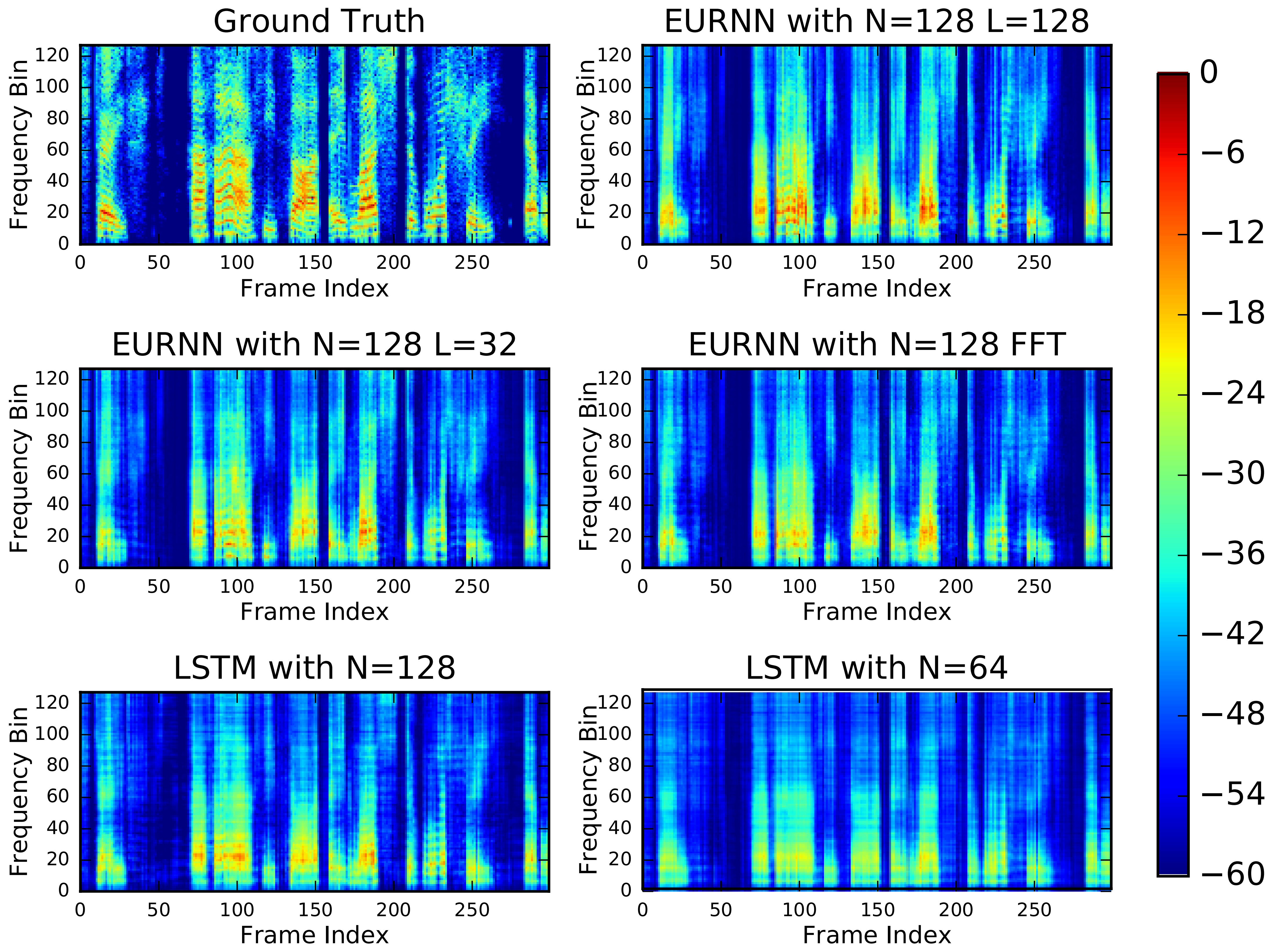
EURNN outperforms LSTM in both final performance and training speed.

EURNN tunable-style achieves highest accuracy with least number of parameters.

Model	hidden size (capacity)	number of parameters	validation accuracy	test accuracy
LSTM	80	16k	0.908	0.902
URNN	512	16k	**0.942**	0.933
PURNN	116	16k	0.922	0.921
EURNN (tunable style)	1024 (2)	13.3k	0.940	**0.937**
EURNN (FFT style)	512 (FFT)	9.0k	0.928	0.925

Experiment results

- Speech spectrum prediction



TIMIT spectrum sampled in 8 GHz. RNN model is required to predict next frame based on previous frames.

EURNN outperforms LSTM with less number of parameters

Model	hidden size (capacity)	number of parameters	MSE (validation)	MSE (test)
LSTM	64	33k	71.4	66.0
LSTM	128	98k	55.3	54.5
EURNN (tunable style)	128 (2)	33k	63.3	63.3
EURNN (tunable style)	128 (32)	35k	52.3	52.7
EURNN (tunable style)	128 (128)	41k	**51.8**	**51.9**
EURNN (FFT style)	128 (FFT)	34k	52.3	52.4

Conclusion

Efficient Unitary NN (EUNN)

- **Efficient**: O(1) operation per parameter
- **Strict**: strictly unitary parametrization
- **Tunable**: from small subspace to full unitary space, trade off capacity to hidden size
- **Easy implementation**: element-wise functions, no need to implement backpropagation
- **FFT approximation**: provides further speed-up

Future work

- Combine unitary matrices with other mechanisms: gated system, attention mechanism etc

- Go beyond Recurrent Neural Network: Parametrization of Semi-unitary Matrices

Thank you!