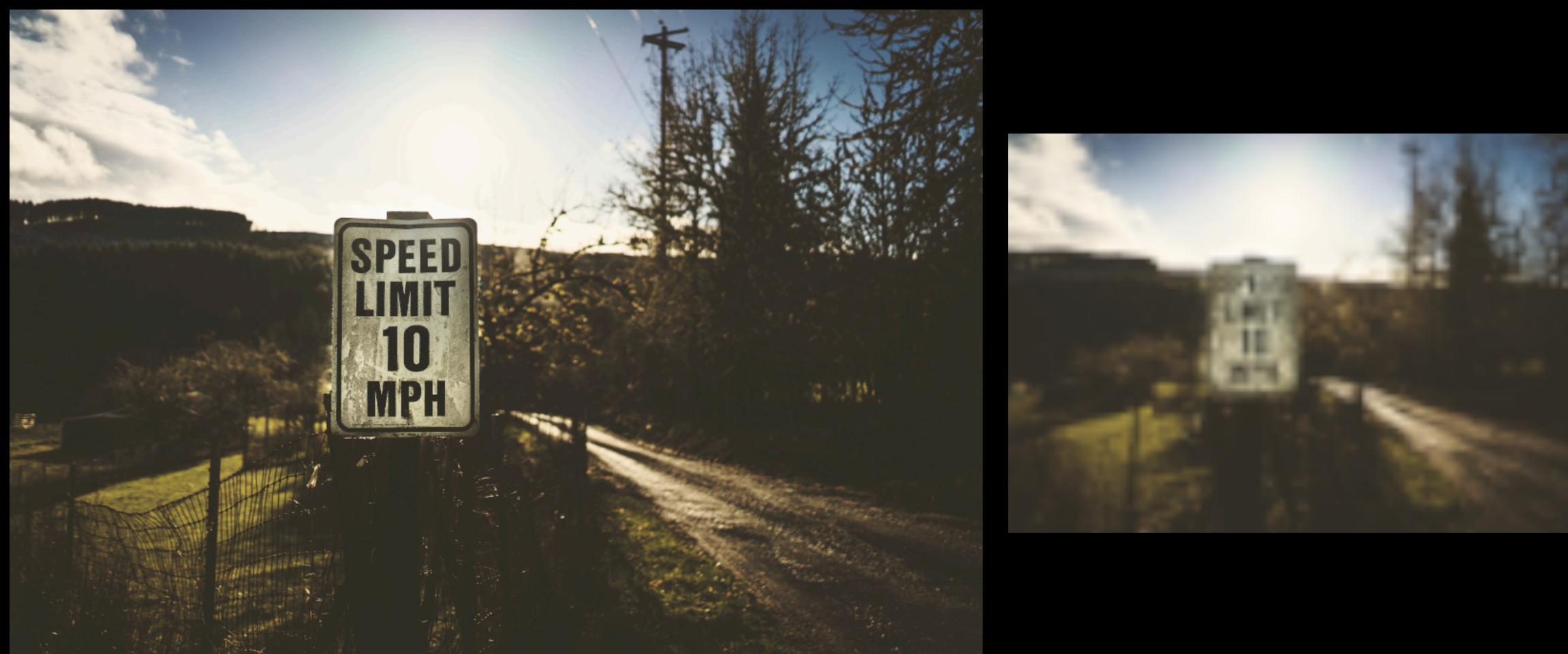


# Variational Saccading: Efficient Inference for Large Resolution Images

Jason Ramapuram, Maurits Diephuis, Frantzeska Lavda, Russ Webb, Alexandros Kalousis  
Jason.Ramapuram@etu.unige.ch



- ### Leveraging Ultra-Large Resolution Images In Modern Neural Networks
- SOTA models operate over small images.
  - Information content can be lost with downsampling.



### We propose

Variational Saccading: an **end-to-end** method that uses a proxy distribution (such as a downsampled image) to learn where to look in the original (large) input distribution.

### Contributions

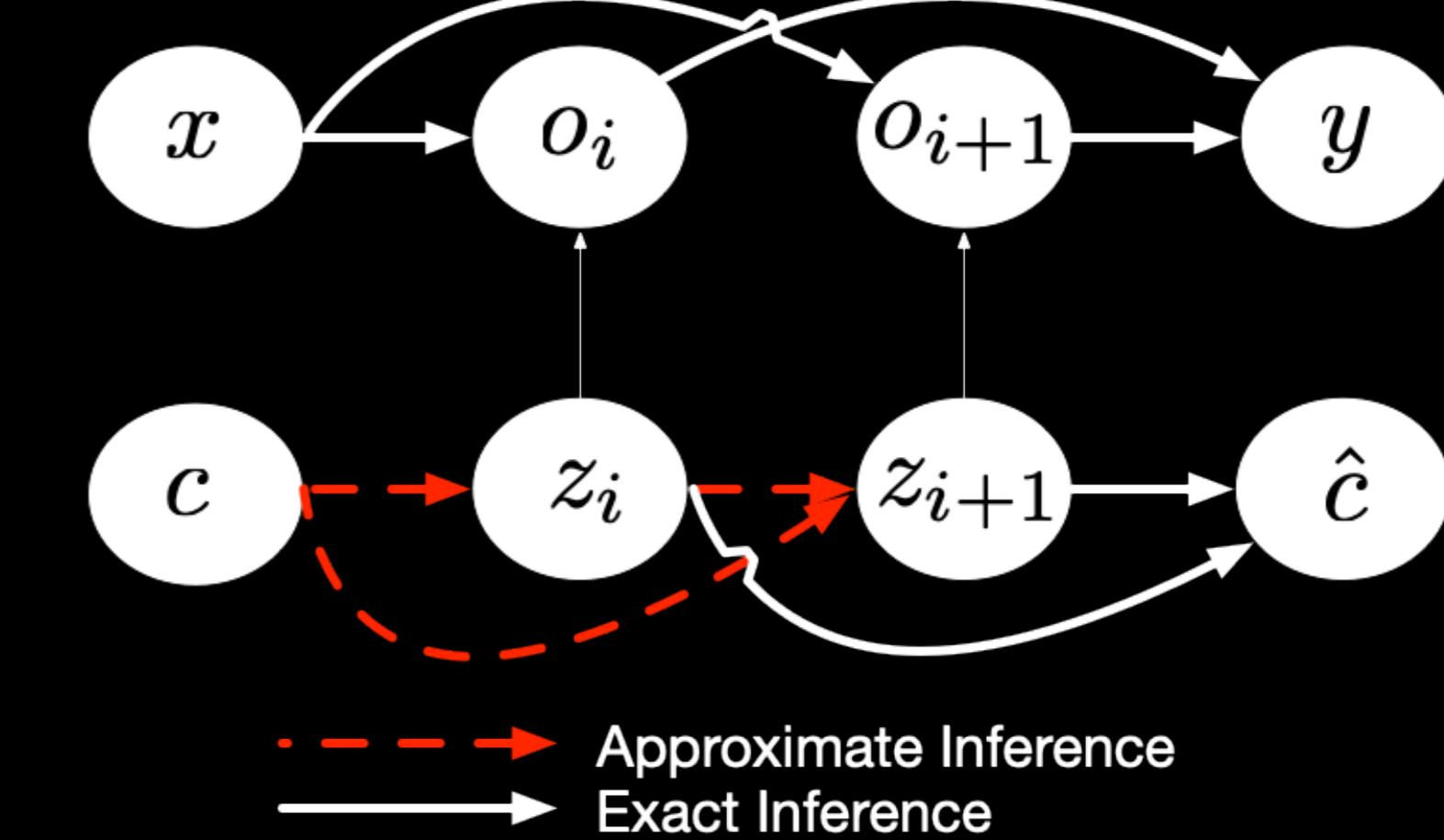
- **10x speedup** in training time per epoch.
- **5% memory usage** compared to traditional approaches.
- **Interpretable crop regions.**
- **End-to-end differentiable** via pathwise estimators.

**Variational Lower Bound Formulation**

$$\text{classification} \mid \text{image} \gtrsim \text{classification} \mid \text{crops} + \text{VRNN ELBO}$$

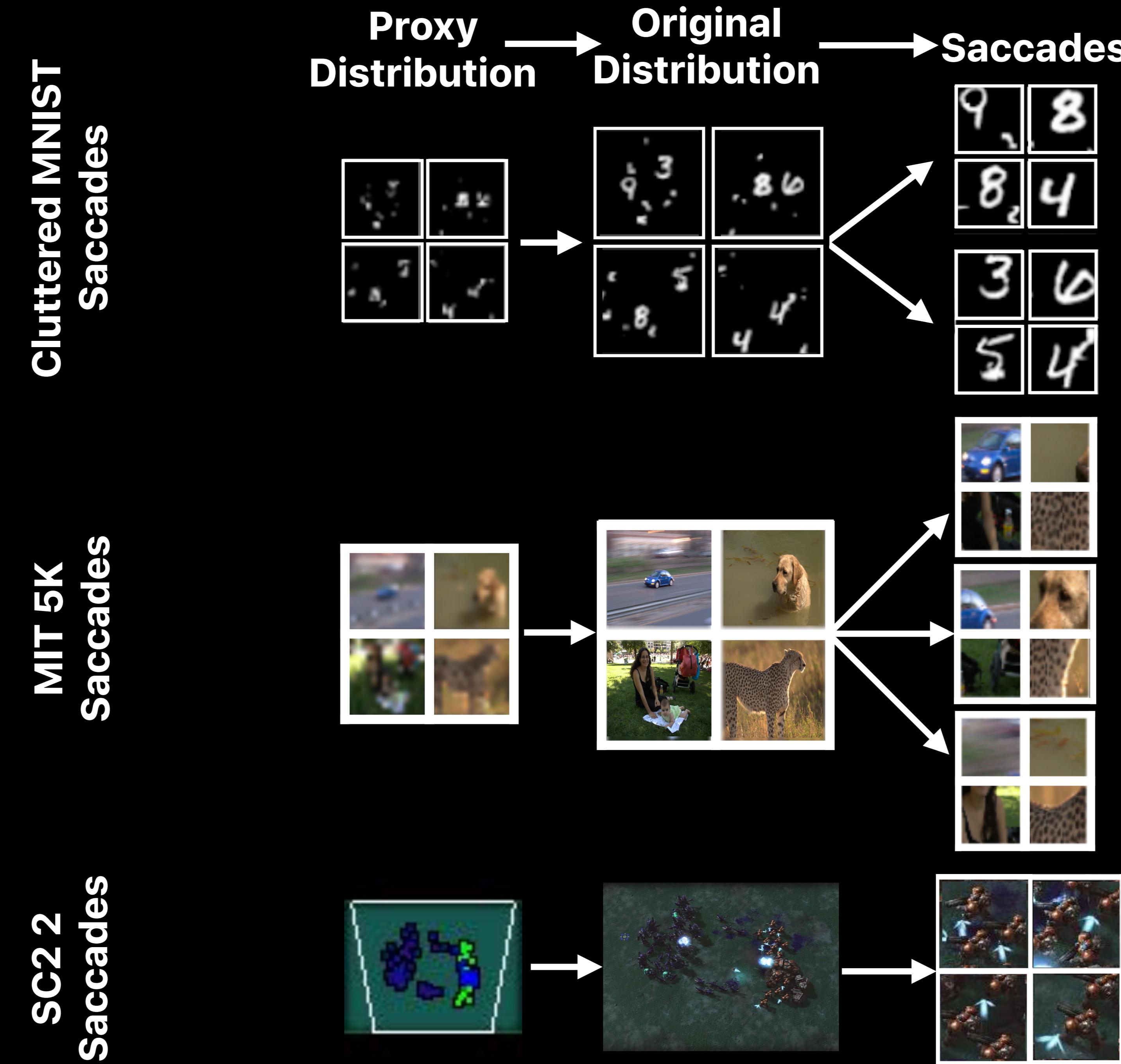
$$\log p_{\theta}(y \mid x) \gtrsim \mathbb{E}_{q_{\phi}} \left( \log p_{\theta_y}(y \mid f_{\theta_{\text{conv}}}(\underbrace{f_{ST}(z_1, h_1, x), \dots, f_{ST}(z_T, h_T, x))}_{\text{crop}_1}, \dots, \underbrace{f_{ST}(z_{T'}, h_{T'}, x))}_{\text{crop}_{T'}}) \right)$$

$$+ \mathbb{E}_{q_{\phi}} \left( \log p_{\theta_c}(\hat{c} \mid z_{\leq T}) - D_{KL}[q_{\phi}(z_{\leq T} \mid z_{< T}, c) \parallel p_{\theta_c}(z_{\leq T} \mid z_{< T}, c)] \right)$$

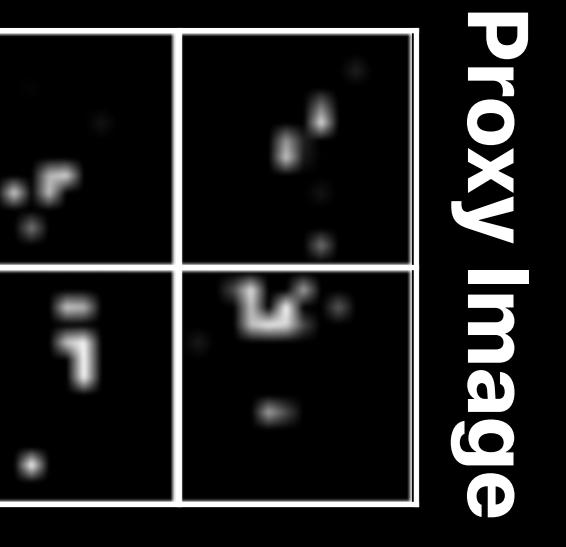
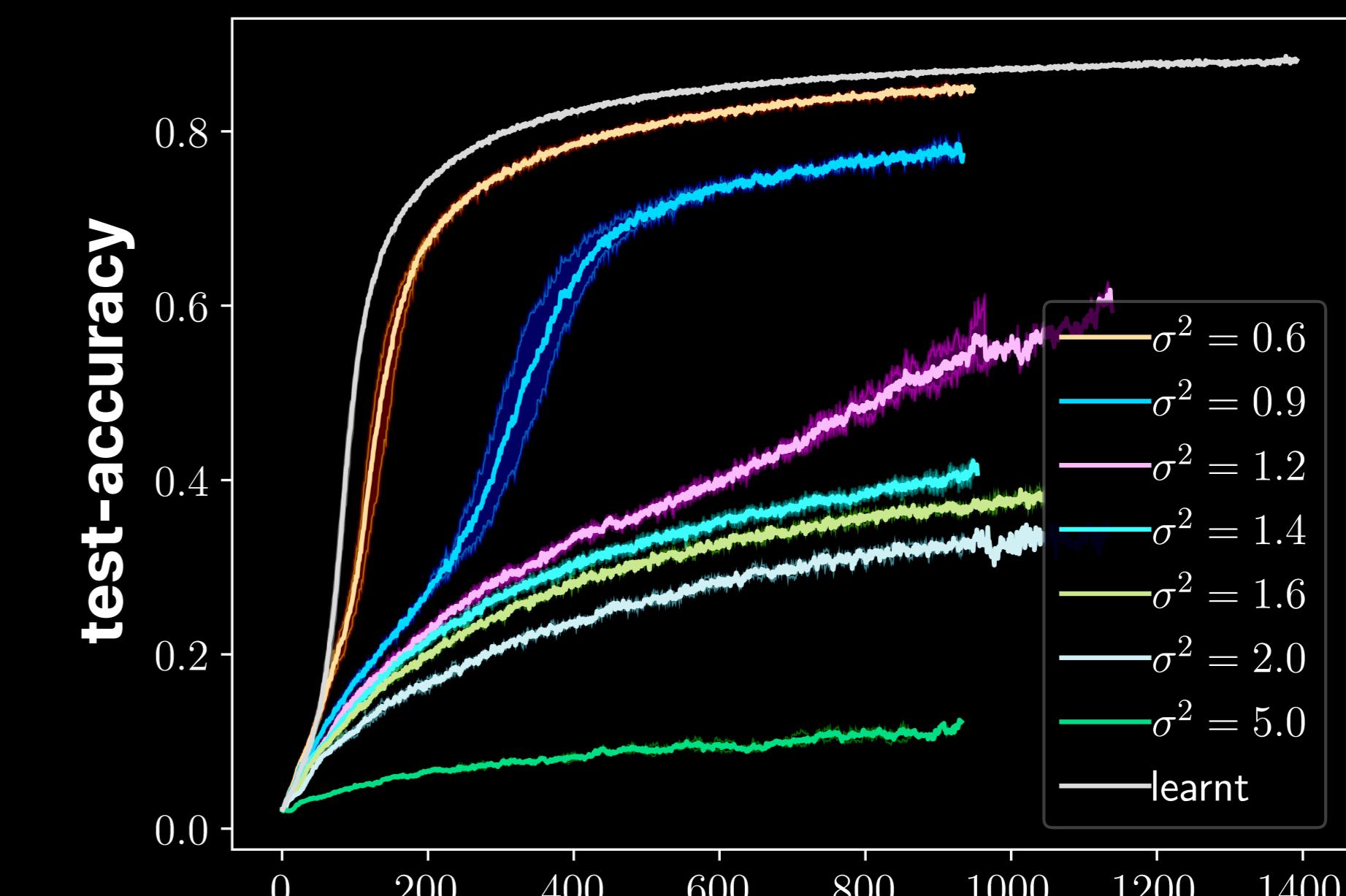


size: 2528×2528	#params	gpu mem	time/epoch	MIT-Adobe-5k	2-MNIST ID
resnet18 (144 crops)	11.4M	79G	1052s	63.6%	97.3%
variational saccading	7.4M	4.1G	120s	62.7%	95.2%

size: 100×100	#params	gpu mem	time/epoch	2-MNIST Sum	2-MNIST ID
resnet18 (full image)	11M	6.6G	59.3s	99.86%	97.4%
RAM (Mnih, 2014)	-	-	-	91%	93%
DRAM (Ba, 2015)	-	-	-	97.5%	95%
variational saccading	7.4M	2.8G	37s	97.2%	95.4%



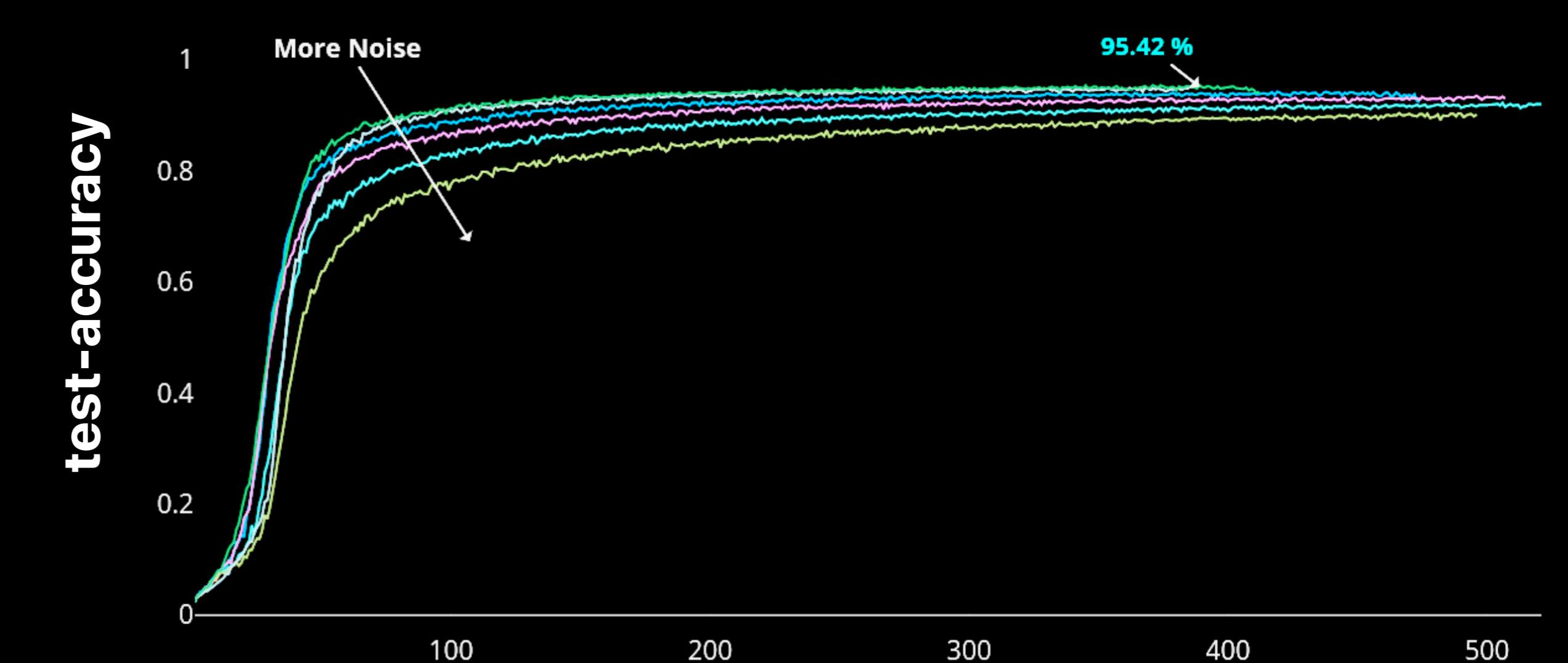
**Learnt vs. Fixed Variance ClutteredMNIST  $\mathbb{R}^{2528 \times 2528}$**



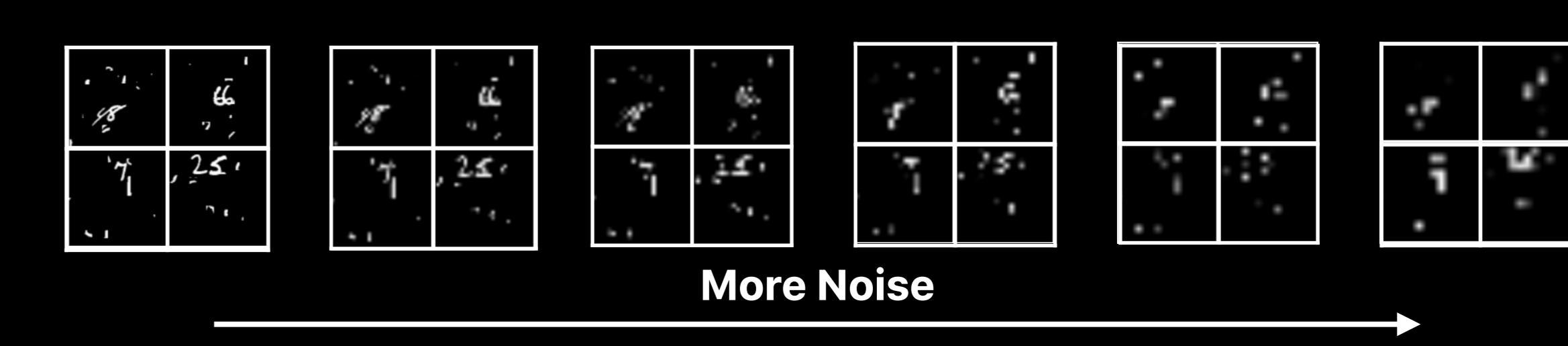
Proxy Image

Real Image

**Effect of Noisy Proxy Distribution**



Epoch



More Noise