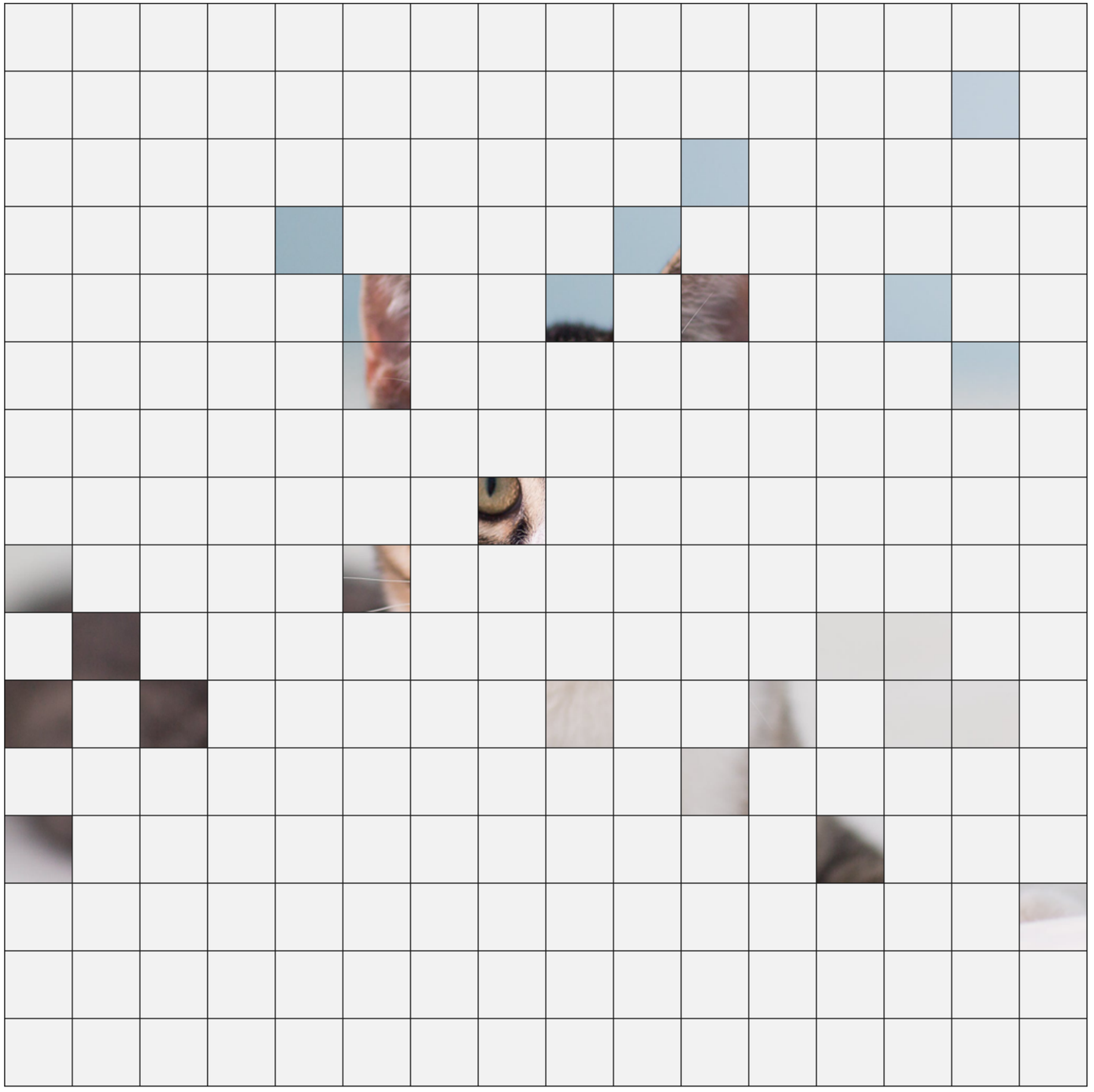
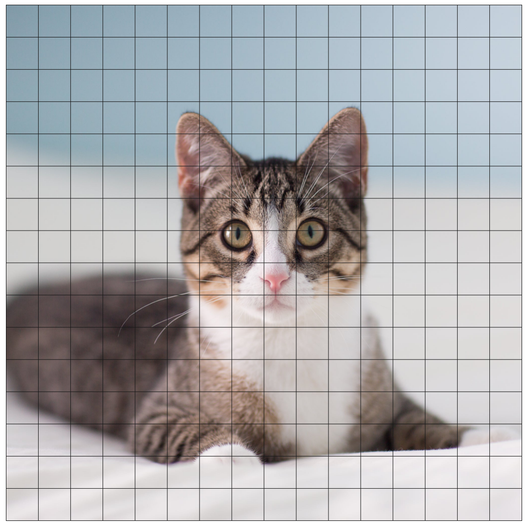
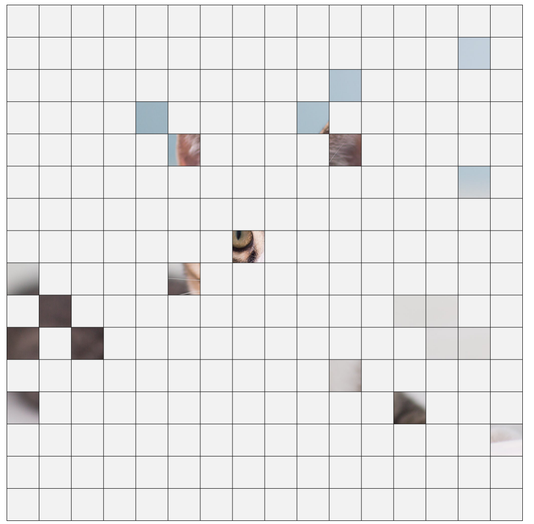
PixelTransformer: Sample Conditioned Signal Generation

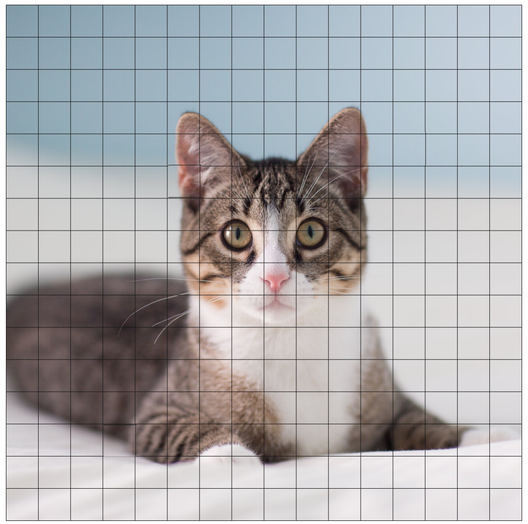
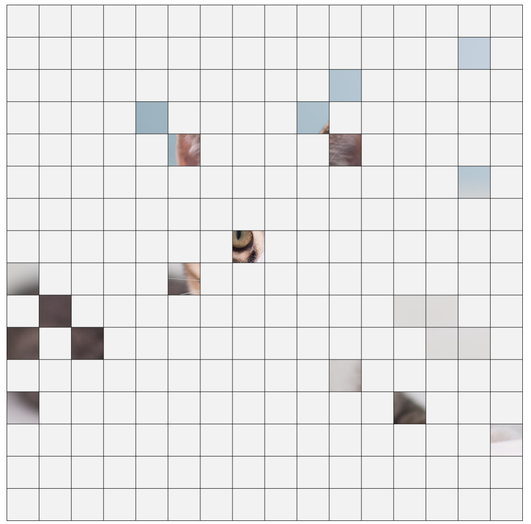
Shubham Tulsiani *(Facebook AI Research)*

Abhinav Gupta *(Facebook AI Research, CMU)*

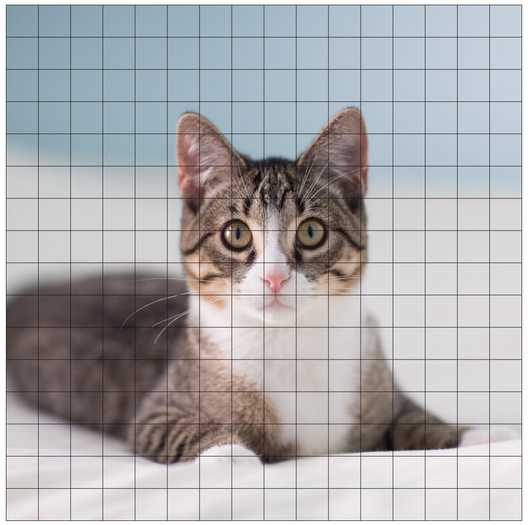
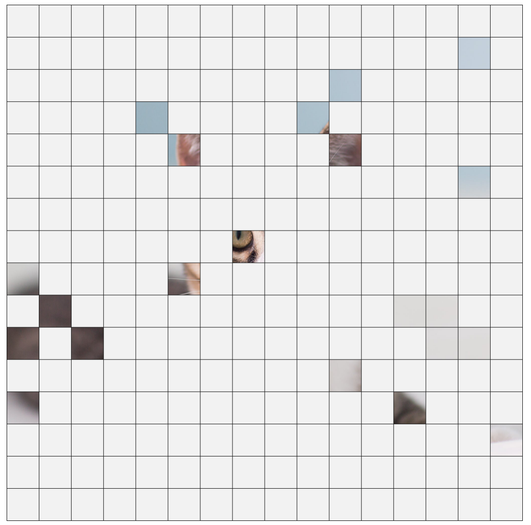


$p($  $|$  $)$

signal distribution conditioned on **samples**

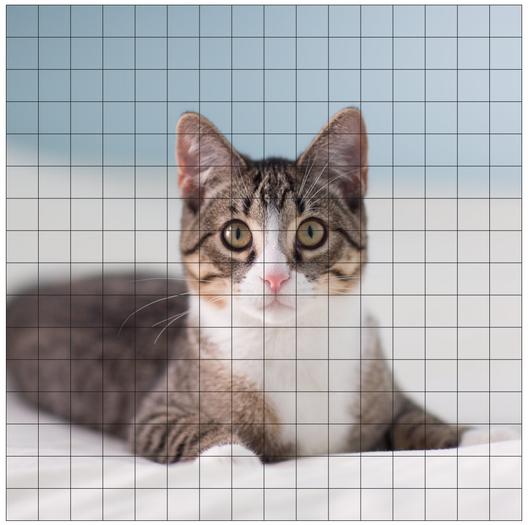
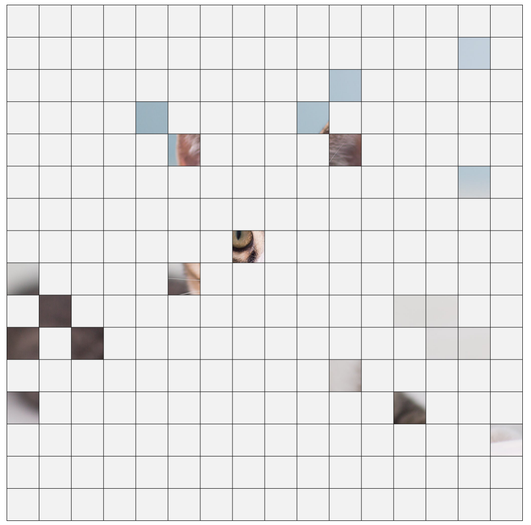
$p($  $|$  $)$

infer: $p(I|S)$

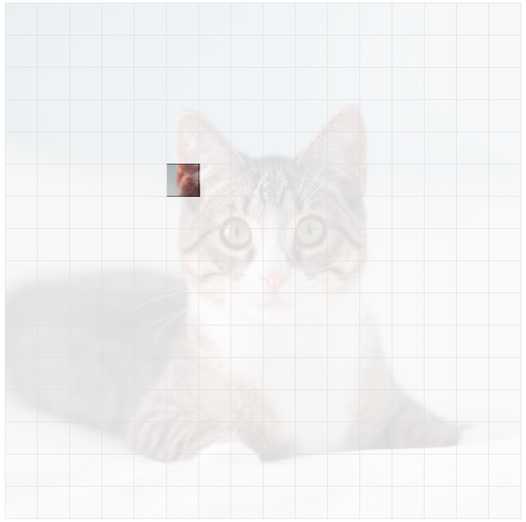
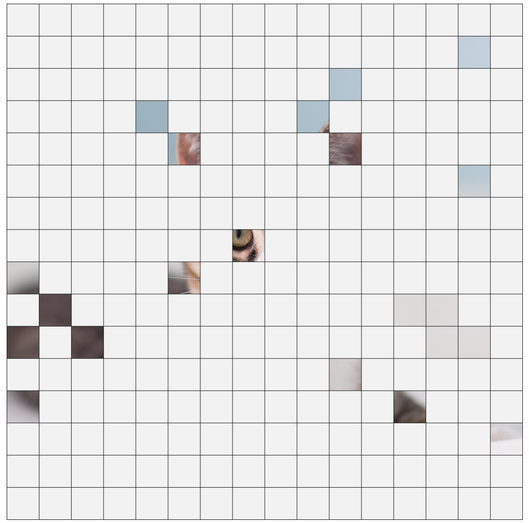
$p($  $|$  $)$

infer: $p(I|S) \equiv p($   ....  $|S)$

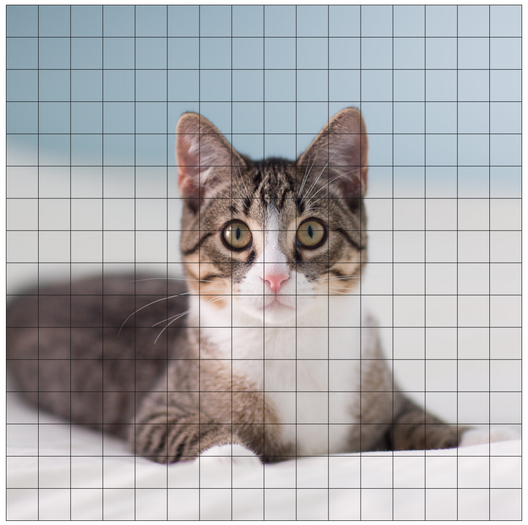
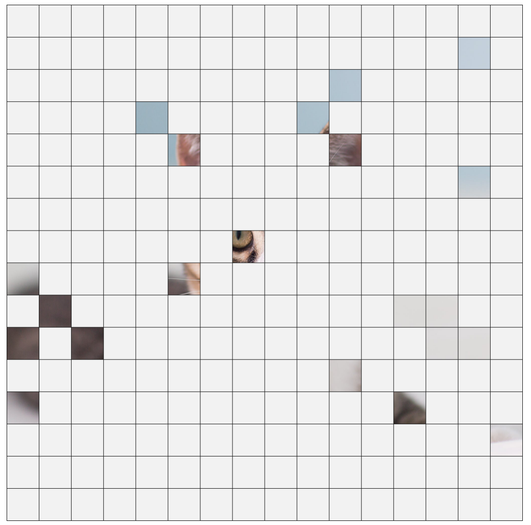
joint distribution over all pixel values

$$p($$  $$\mid$$  $$)$$

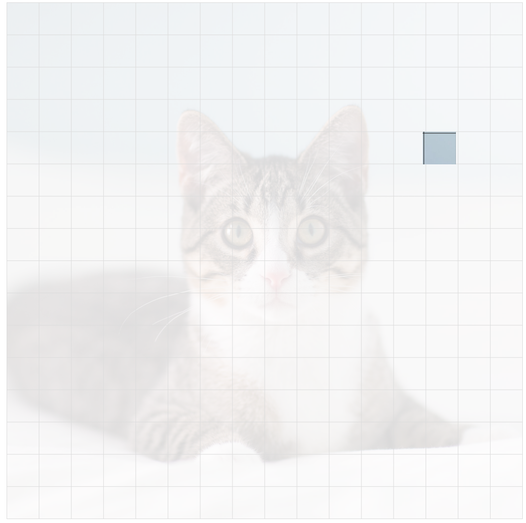
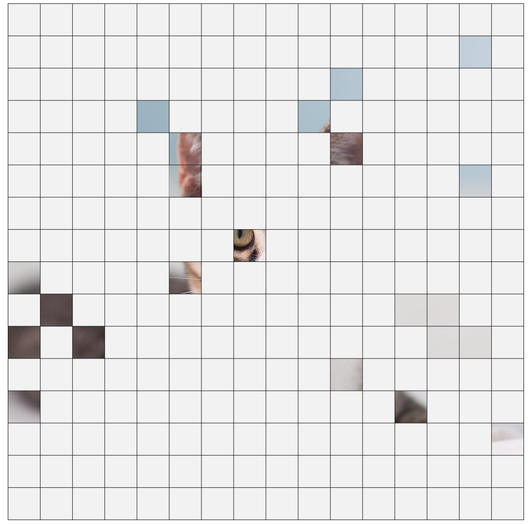
$$\equiv$$

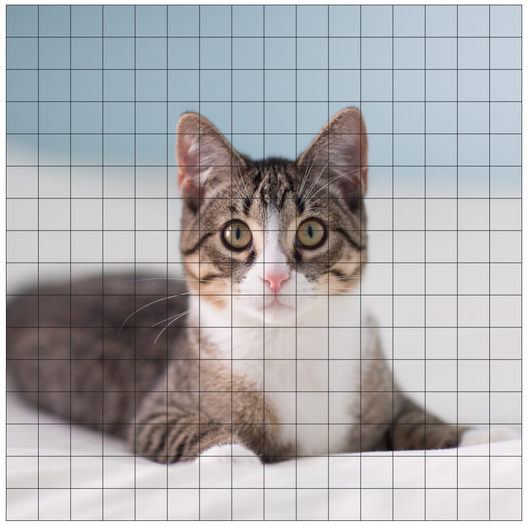
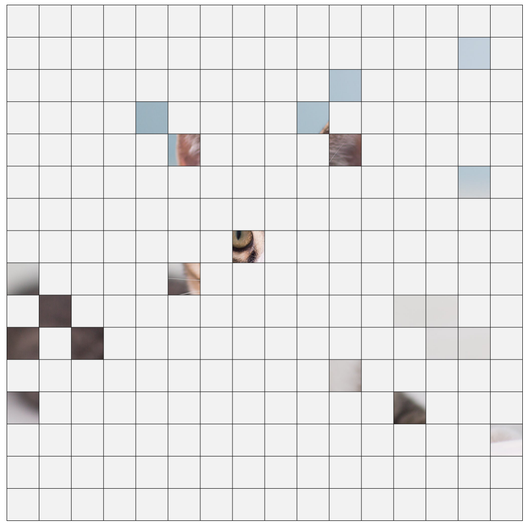
$$p($$  $$\mid$$  $$)$$

$$p(\mathbf{v}_{g_1} | S)$$

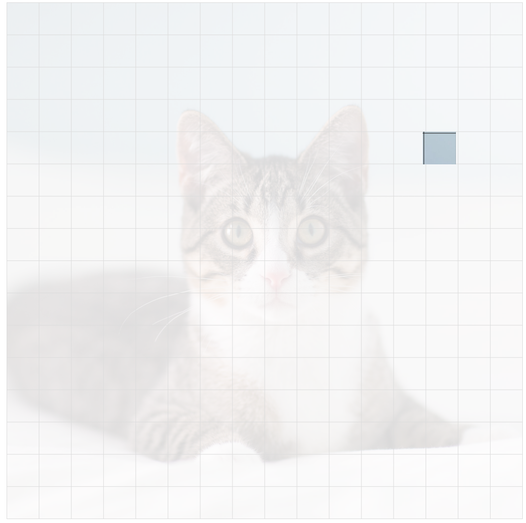
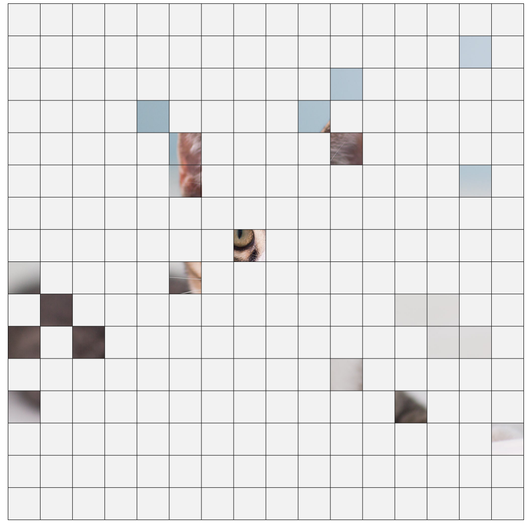
$p($  $|$  $)$

$\equiv$

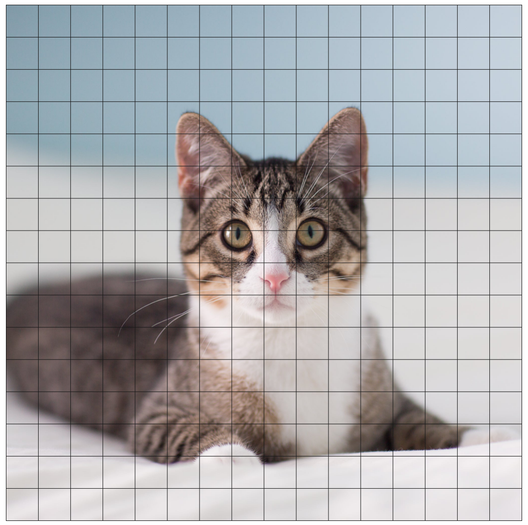
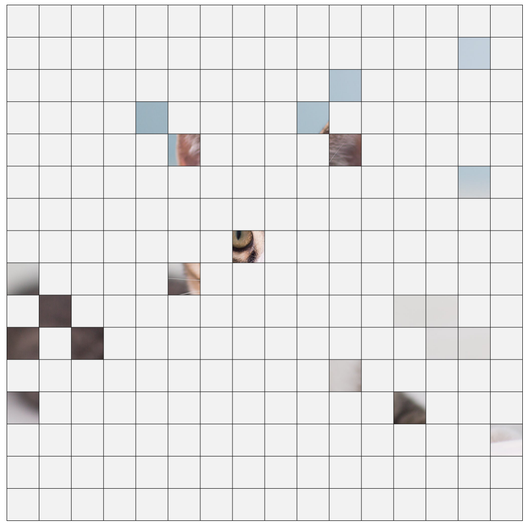
$p(\mathbf{v}_{g_1}|S)$ $p($  $|$  $)$

$p($  $|$  $)$

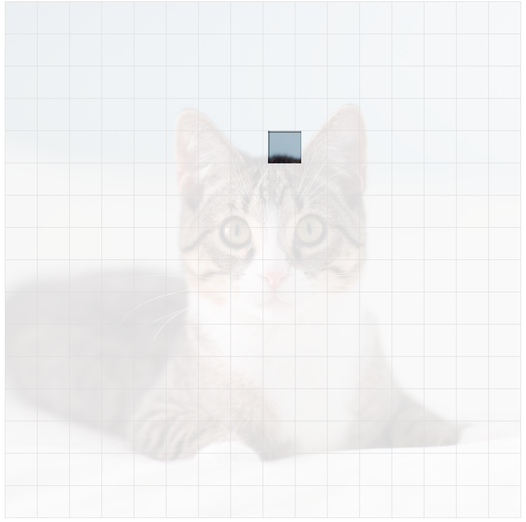
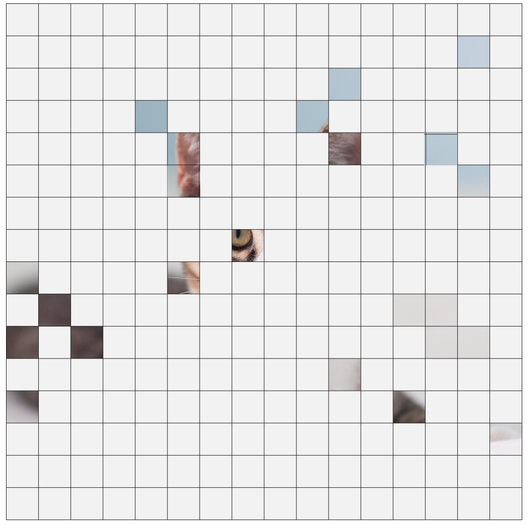
$\equiv$

$p(\mathbf{v}_{g_1}|S)$ $p($  $|$  $)$

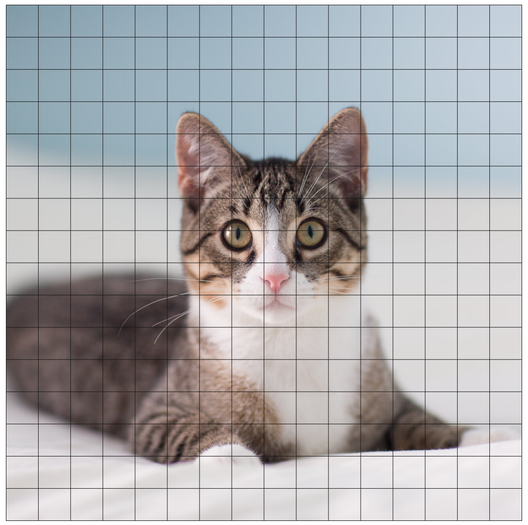
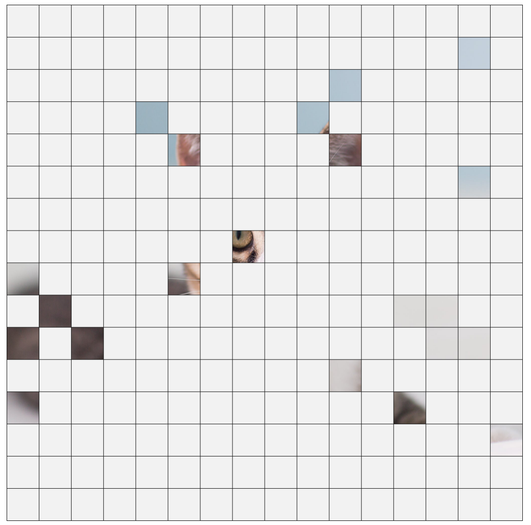
$p(\mathbf{v}_{g_2}|S_1)$

$$p($$  $$|$$  $$)$$

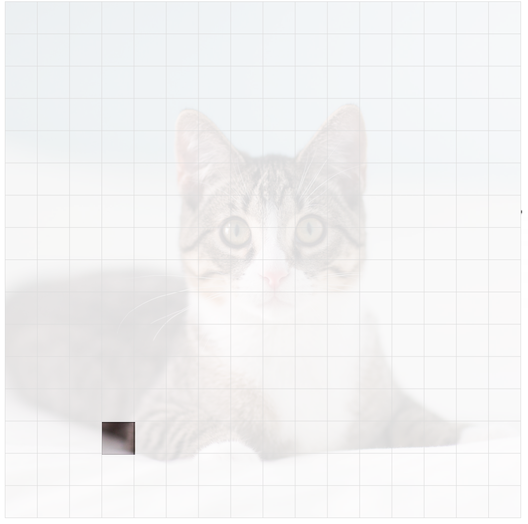
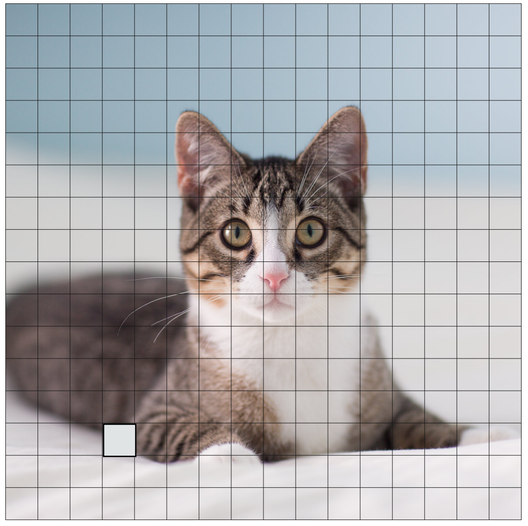
$$\equiv$$

$$p(\mathbf{v}_{g_1}|S) \quad p(\mathbf{v}_{g_2}|S_1) \quad p($$  $$|$$  $$)$$

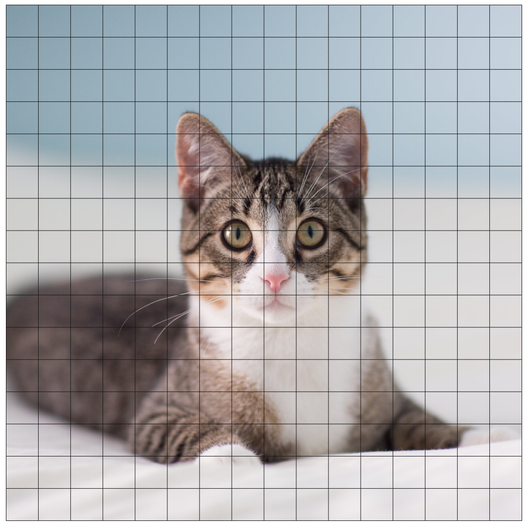
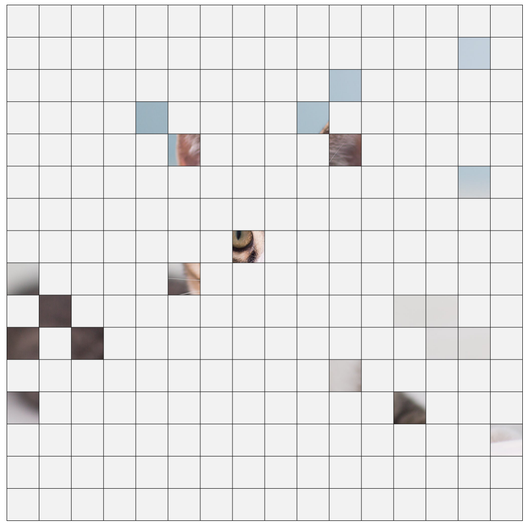
$$p(\mathbf{v}_{g_3}|S_2)$$

$$p($$  $$\Big|$$  $$)$$

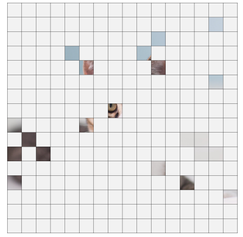
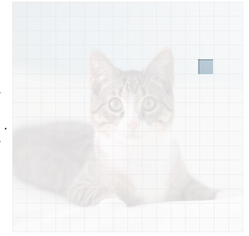
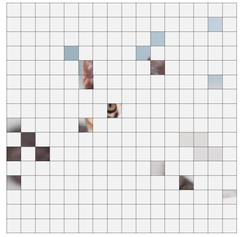
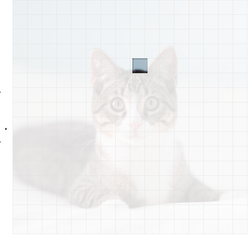
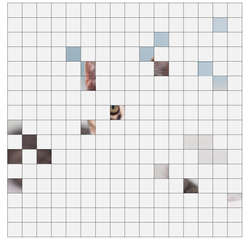
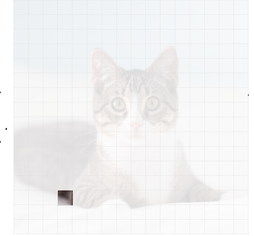
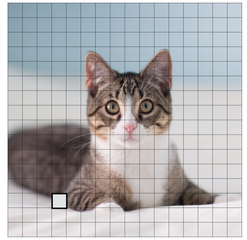
$$\equiv$$

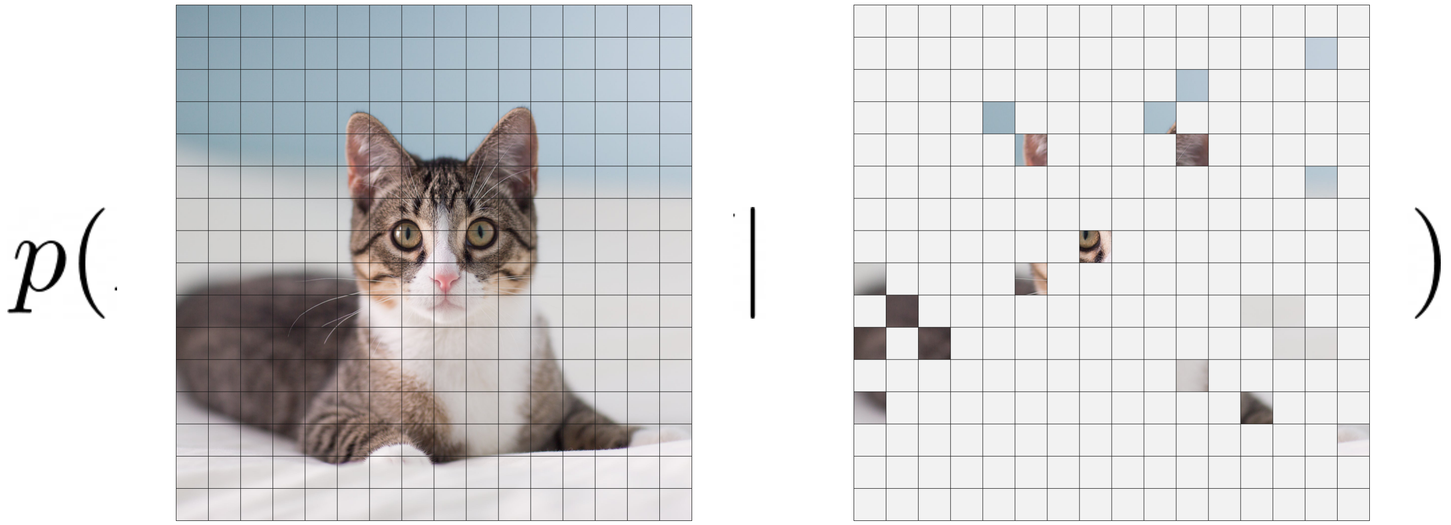
$$p(\mathbf{v}_{g_1}|S) \quad p(\mathbf{v}_{g_2}|S_1) \quad p(\mathbf{v}_{g_3}|S_2) \cdots p($$  $$\Big|$$  $$)$$

$$p(\mathbf{v}_{g_N}|S_{N-1})$$

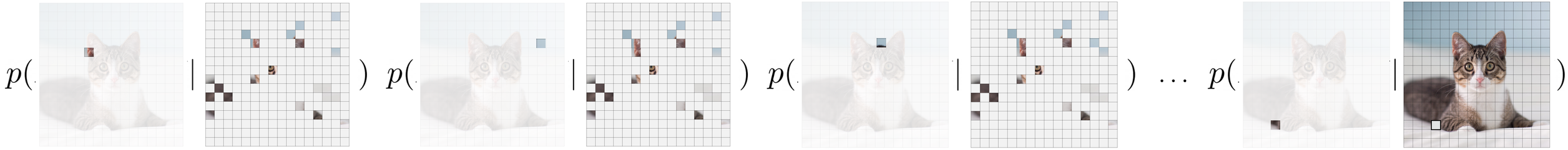
$p($  $|$  $)$

$\equiv$

$p($ $|$  $)$ $p($  $|$  $)$ $p($  $|$  $)$ $\ldots$ $p($  $|$  $)$

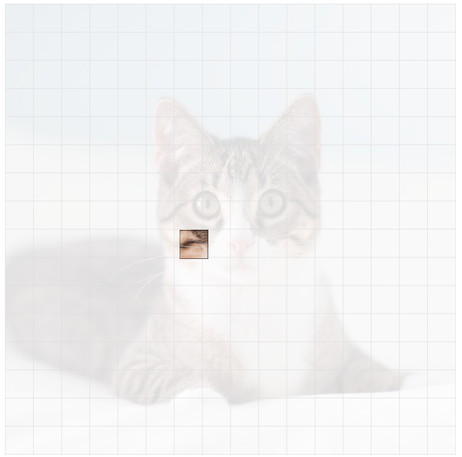
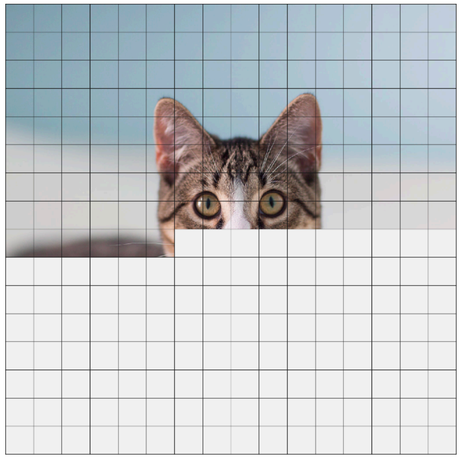


$$\equiv$$



$$p(\mathbf{v_x}|S)$$ is all you need!

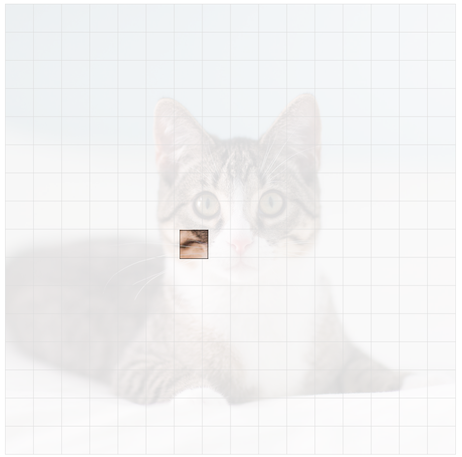
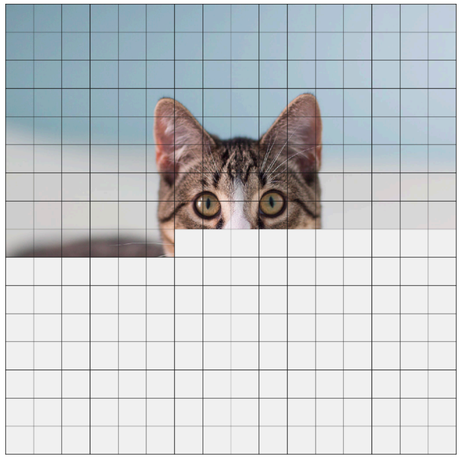
$p(\mathbf{v_x}|S)$ is all you need!

$p($  $|$  $)$

(Sequential) Autoregressive Models

e.g. PixelCNN, PixelRNN, PixelCNN++, Image-GPT

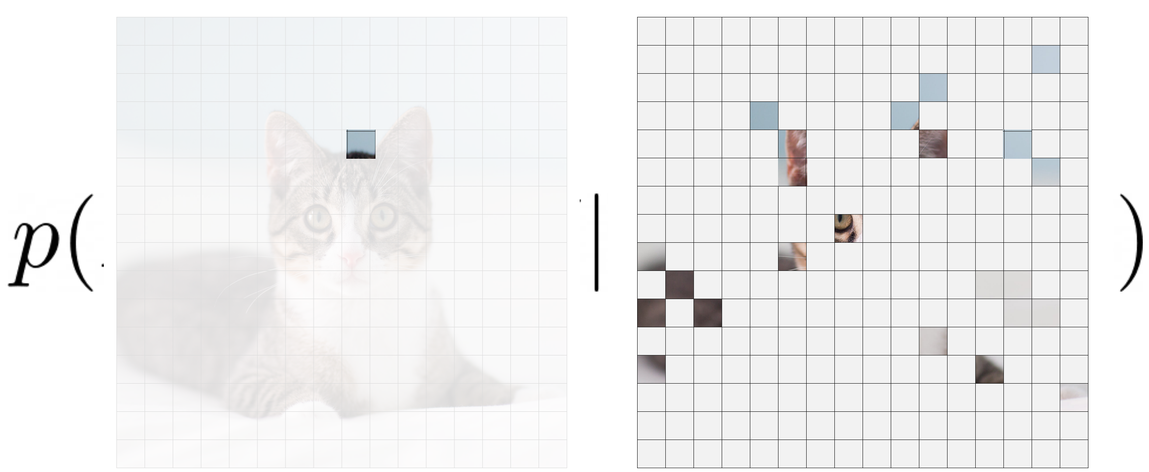
$p(\mathbf{v_x}|S)$ is all you need!

$p($  $|$  $)$

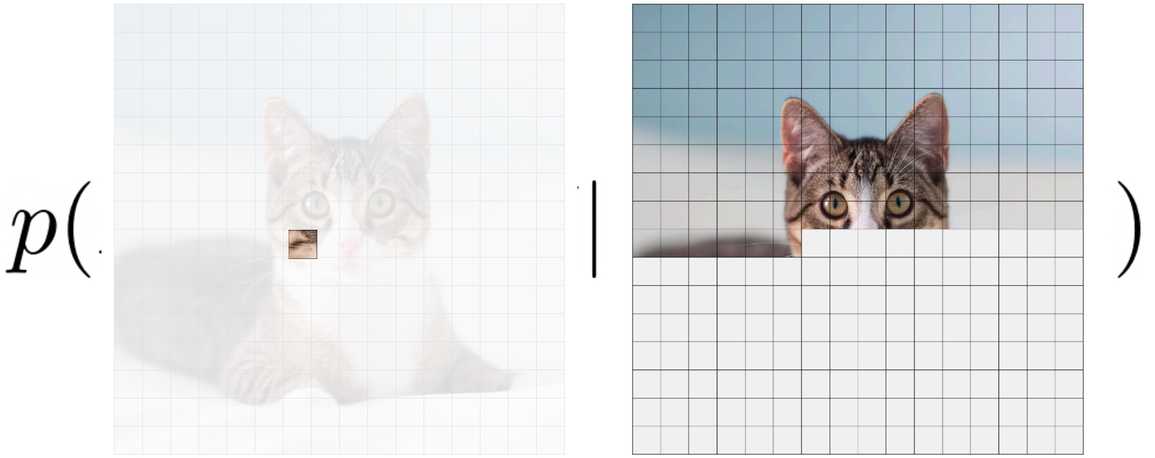
(Sequential) Autoregressive Models
e.g. PixelCNN, PixelRNN, PixelCNN++, Image-GPT

Ordered *S;* x = ‘next’ pixel

$p(\mathbf{v_x}|S)$ is all you need!



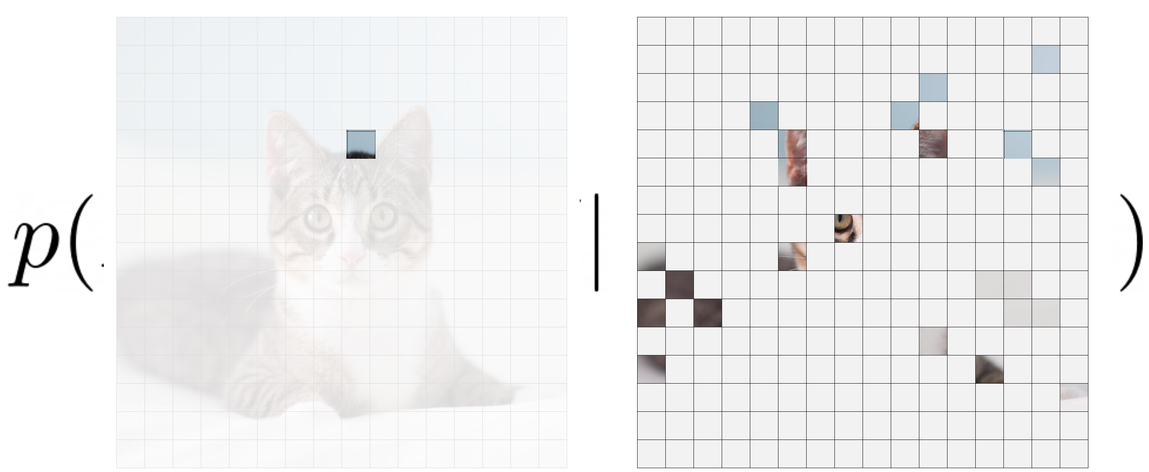
Ours



(Sequential) Autoregressive Models
e.g. PixelCNN, PixelRNN, PixelCNN++, Image-GPT

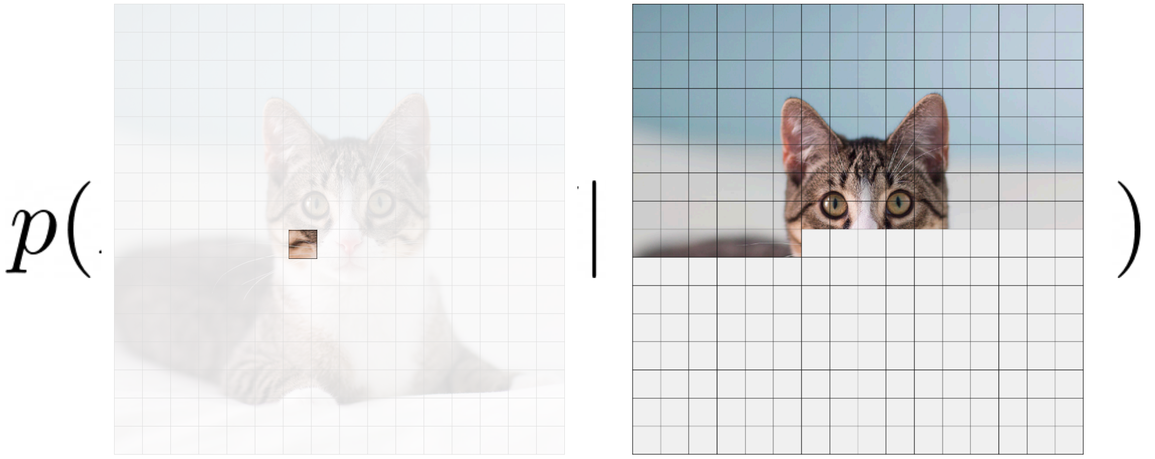
Ordered *S;* x = ‘next’ pixel

$p(\mathbf{v_x}|S)$ is all you need!



Ours

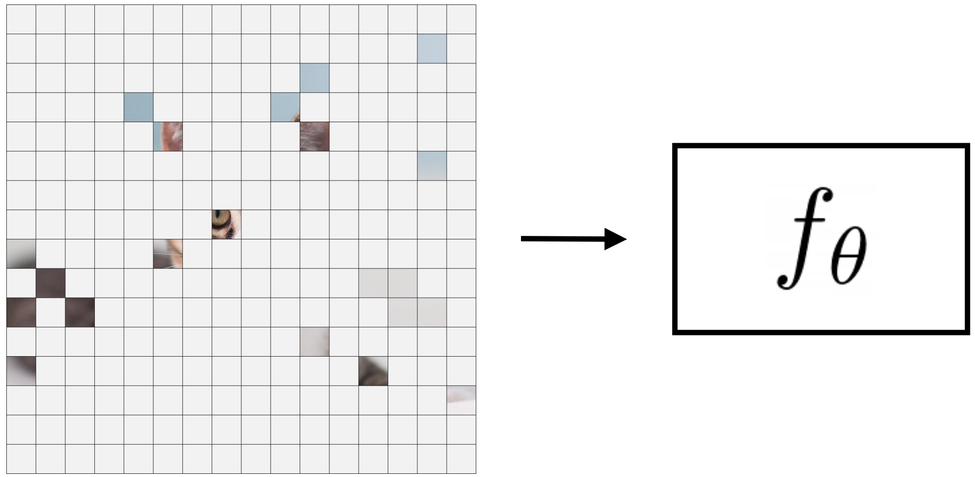
Arbitrary S and x



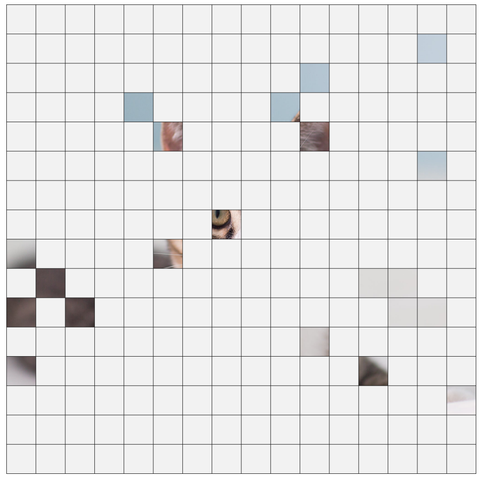
(Sequential) Autoregressive Models
e.g. PixelCNN, PixelRNN, PixelCNN++, Image-GPT

Ordered *S;* x = ‘next’ pixel

$p(\mathbf{v_x}|S)$ is all you need!



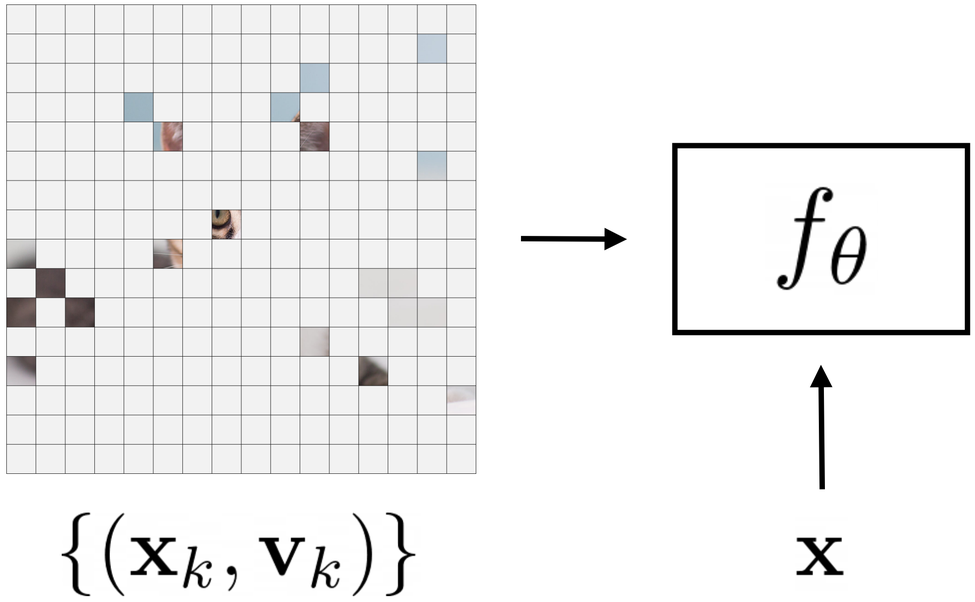
$p(\mathbf{v}_\mathbf{x}|S)$ is all you need!



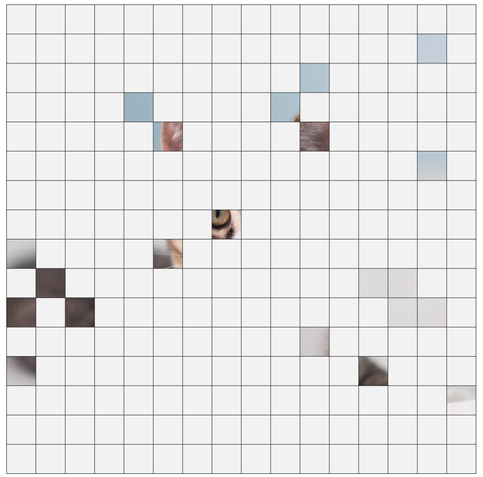
$\{(\mathbf{x}_k, \mathbf{v}_k)\}$

f_θ

$p(\mathbf{v}_\mathbf{x}|S)$ is all you need!



$p(\mathbf{v}_\mathbf{x}|S)$ is all you need!

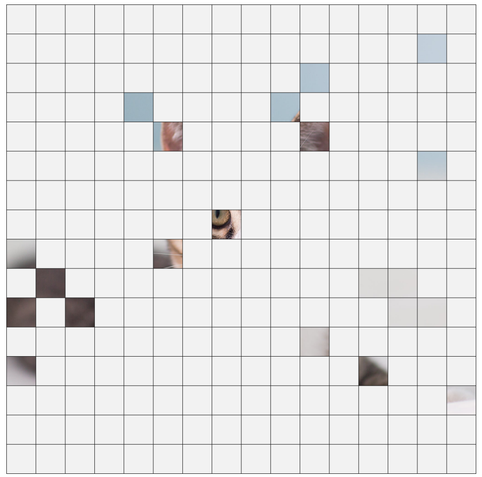


$\{(\mathbf{x}_k, \mathbf{v}_k)\}$

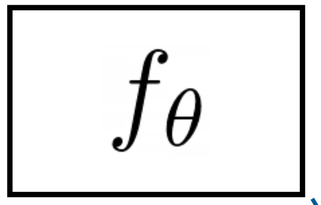
f_θ

$\mathbf{x}$

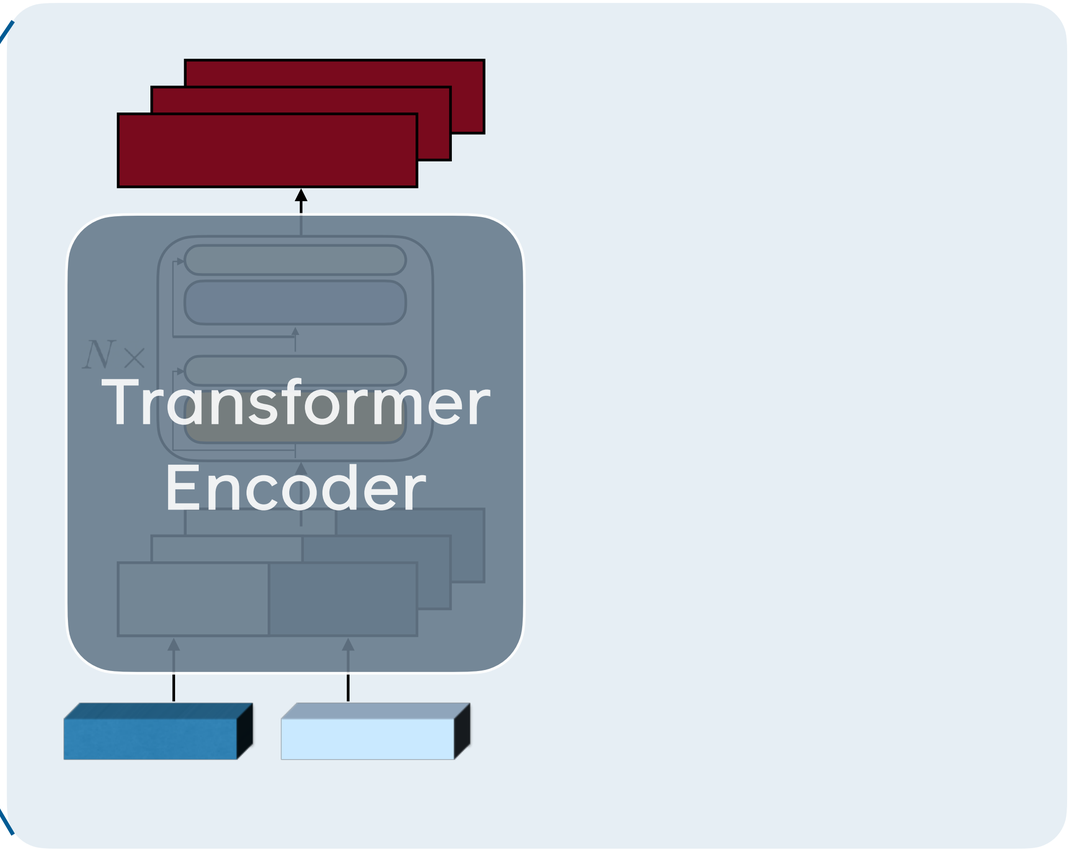
$p(\mathbf{v}_\mathbf{x}|S)$ is all you need!



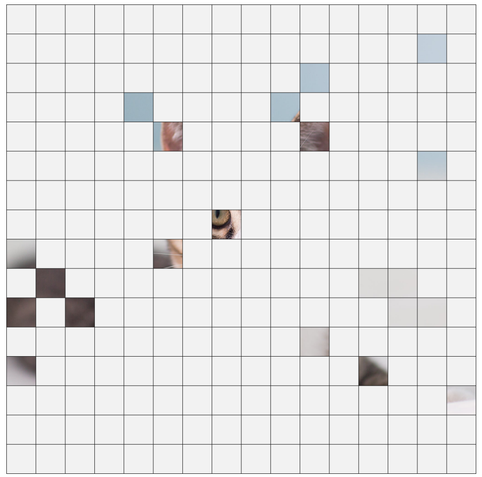
$\{(\mathbf{x}_k, \mathbf{v}_k)\}$



$\mathbf{x}$



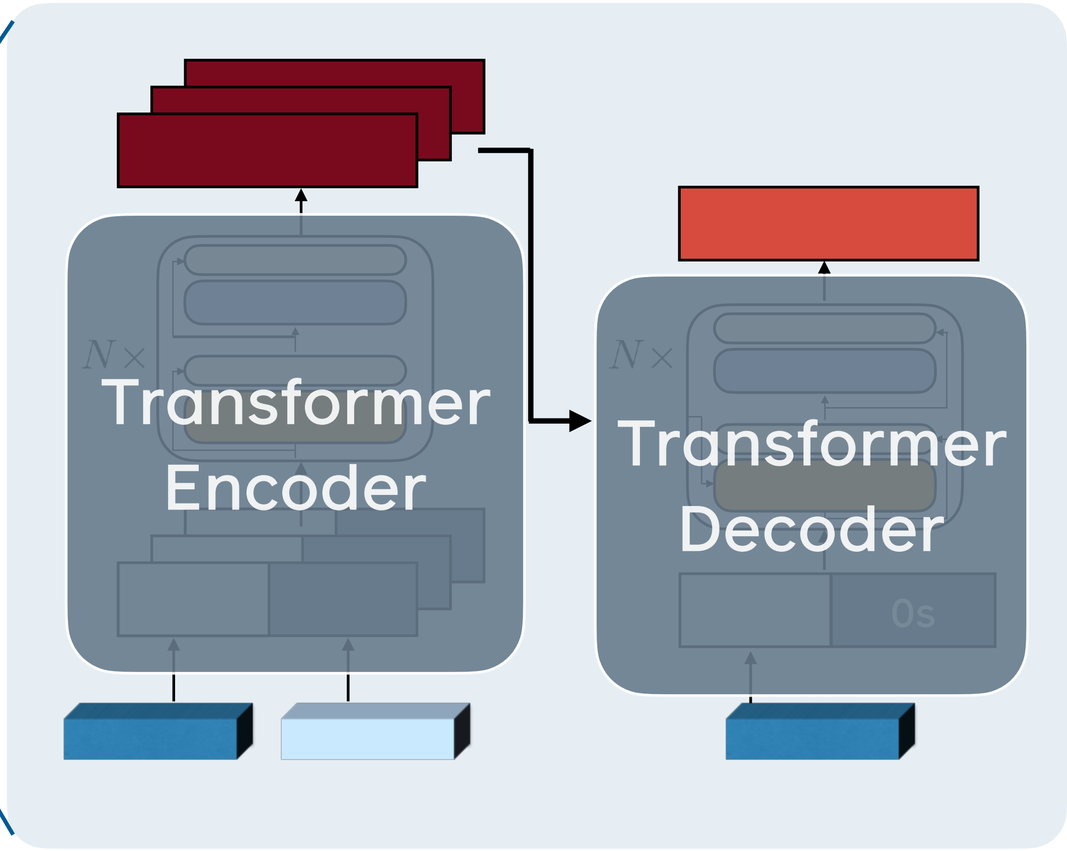
$p(\mathbf{v}_\mathbf{x}|S)$ is all you need!



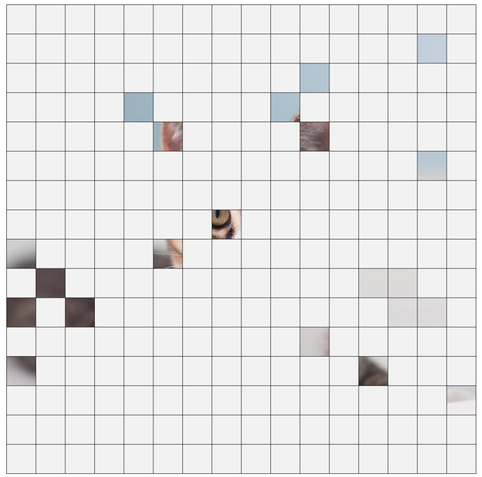
$\{(\mathbf{x}_k, \mathbf{v}_k)\}$

f_θ

$\mathbf{x}$



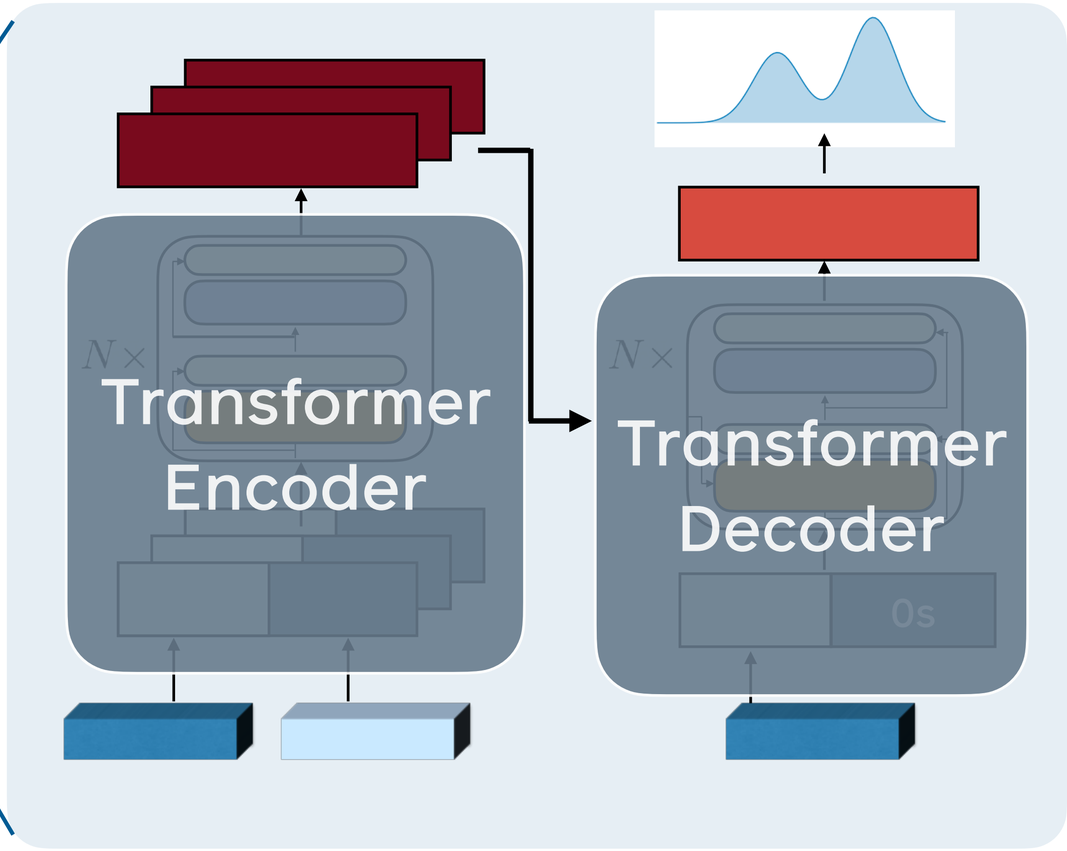
$p(\mathbf{v}_\mathbf{x}|S)$ is all you need!



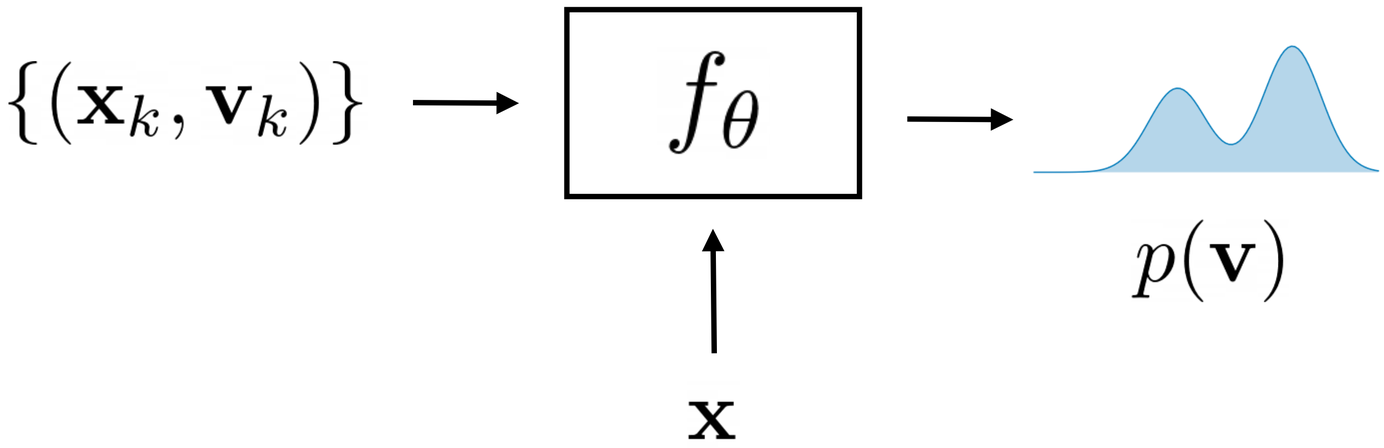
$\{(\mathbf{x}_k, \mathbf{v}_k)\}$

f_θ

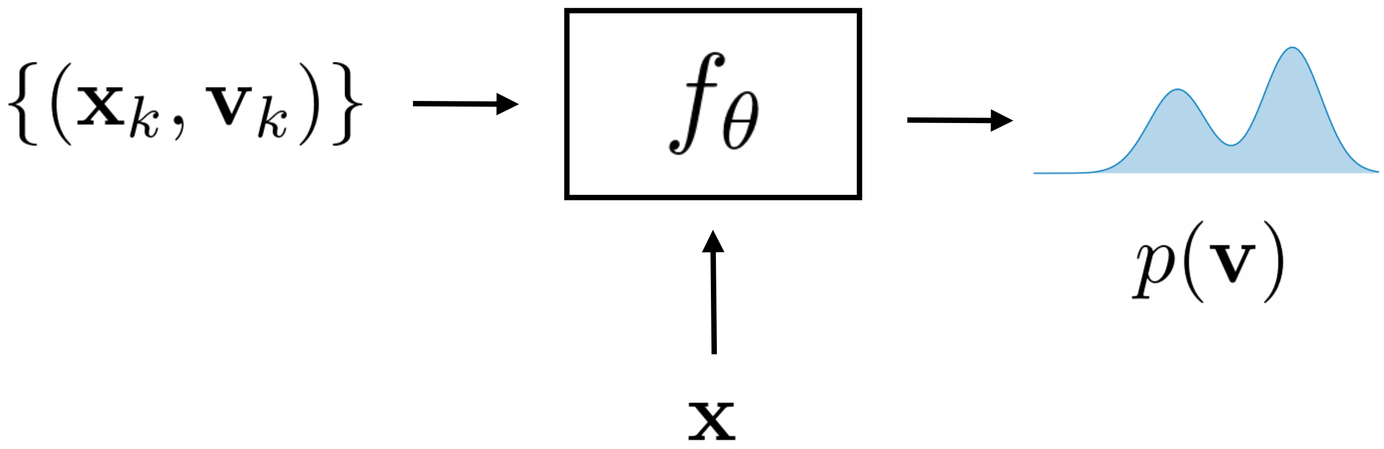
$\mathbf{x}$



$p(\mathbf{v}_\mathbf{x}|S)$ is all you need!

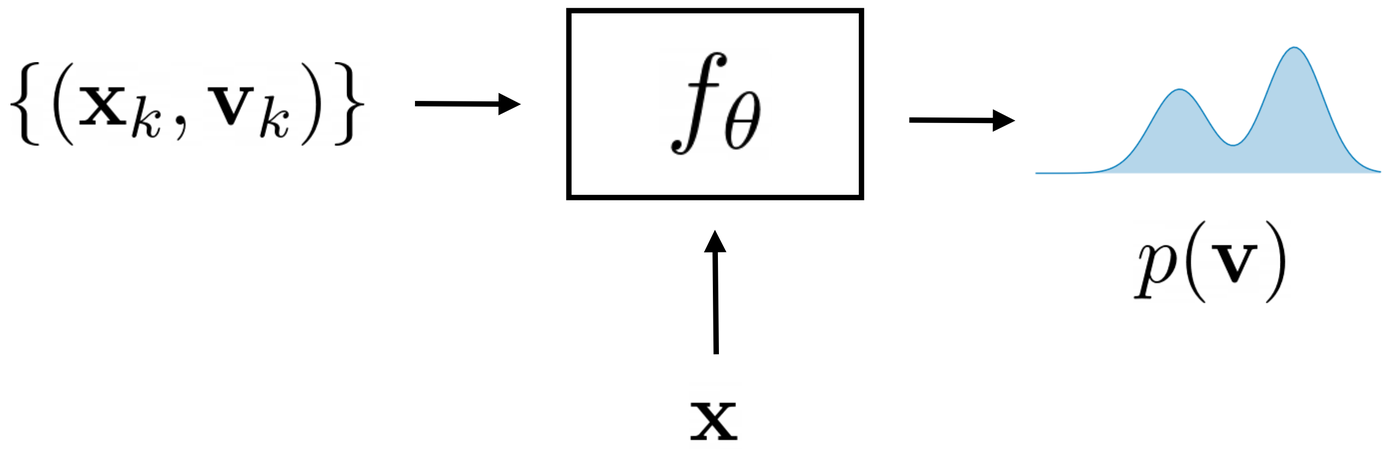


$p(\mathbf{v}_\mathbf{x}|S)$ is all you need!



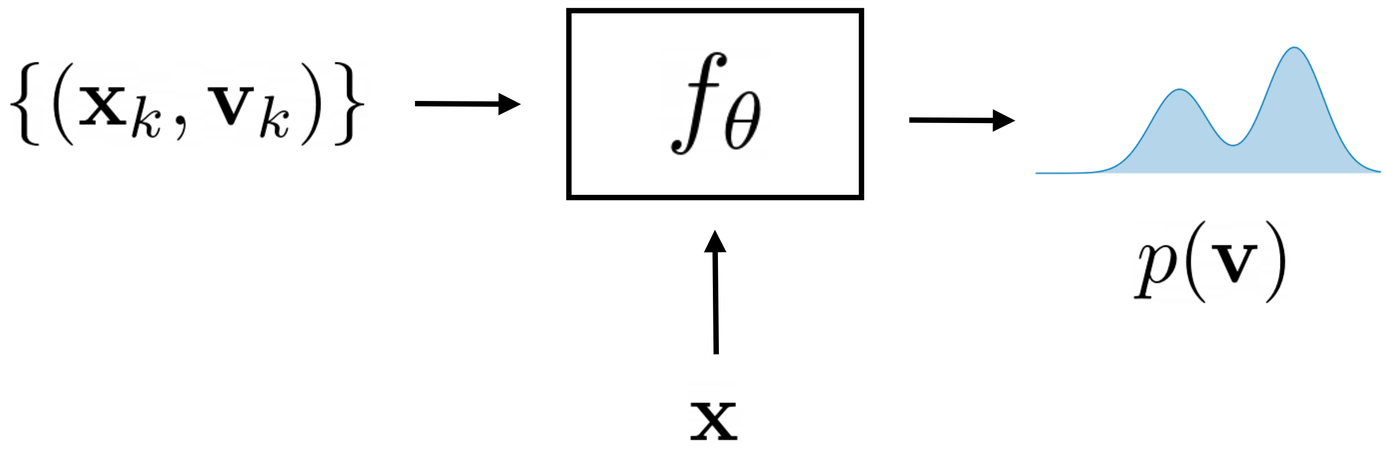
How do we enable learning?

$p(\mathbf{v}_\mathbf{x}|S)$ is all you need!



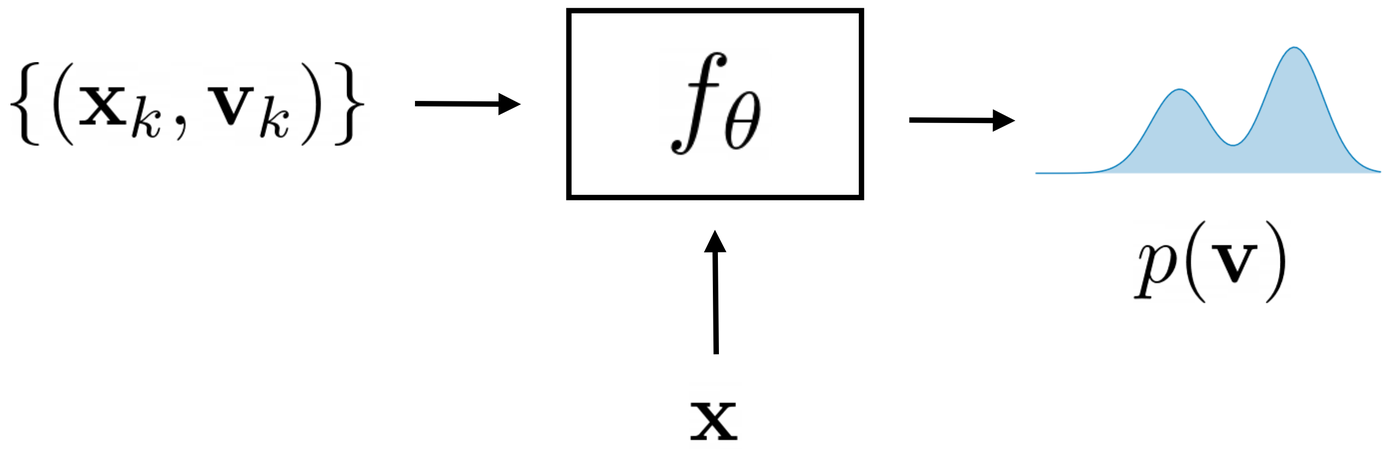
How do we enable learning?

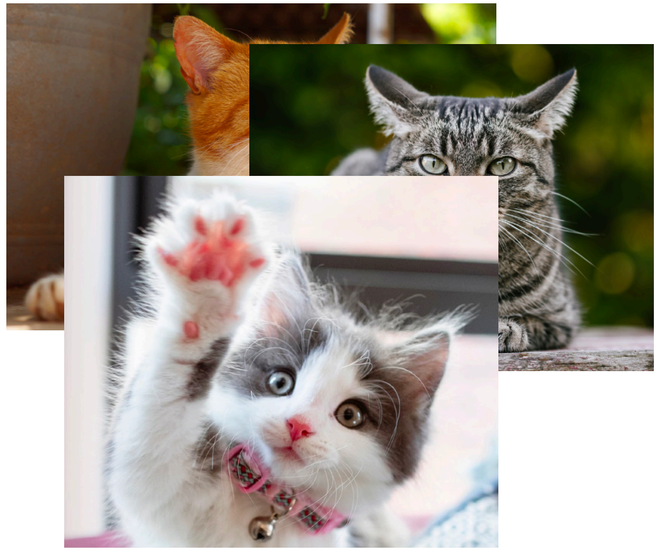
$p(\mathbf{v}_\mathbf{x}|S)$ is all you need!



How do we enable learning?

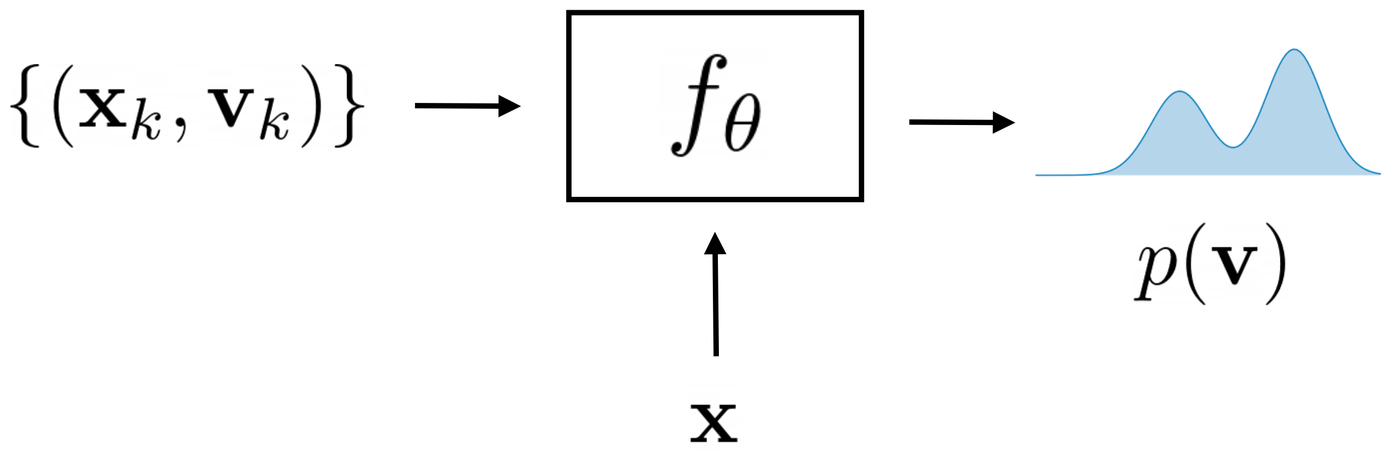
$p(\mathbf{v}_\mathbf{x}|S)$ is all you need!





How do we enable learning?

$p(\mathbf{v}_\mathbf{x}|S)$ is all you need!

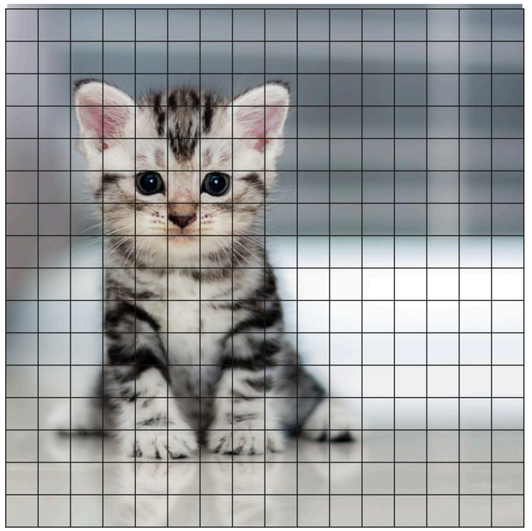


How do we enable learning?

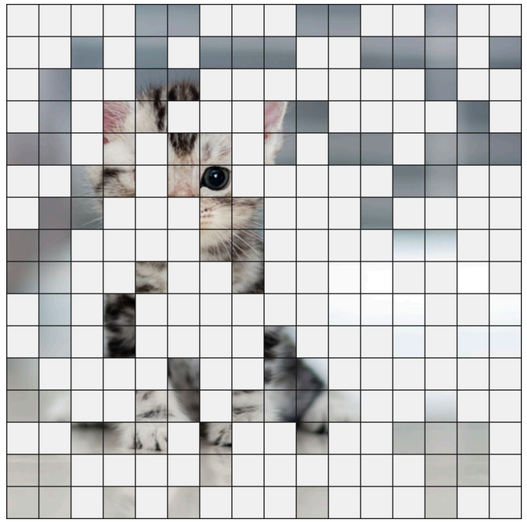
Learning from Self-supervision

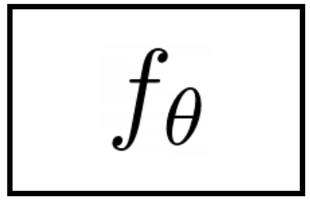
$\{(\mathbf{x}_k, \mathbf{v}_k)\}$ → f_θ → $p(\mathbf{v})$

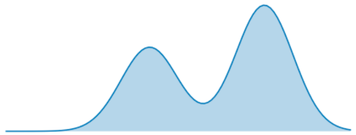
$\mathbf{x}$



Learning from Self-supervision

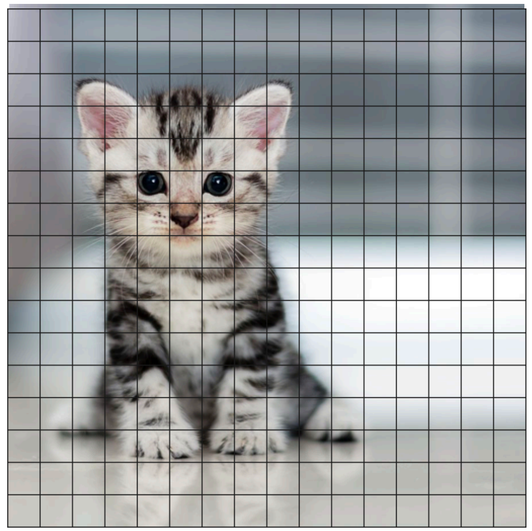




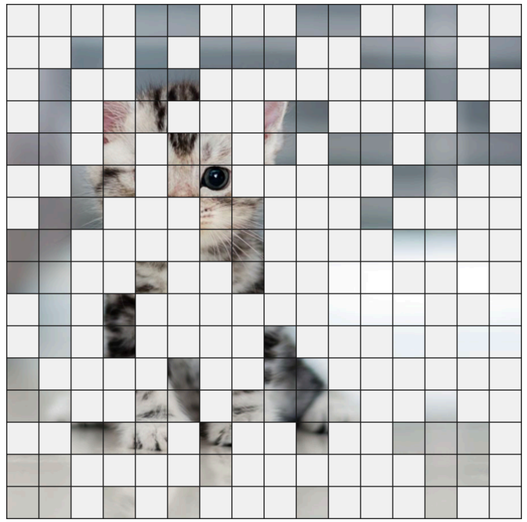


$p(\mathbf{v})$

$\mathbf{x}$



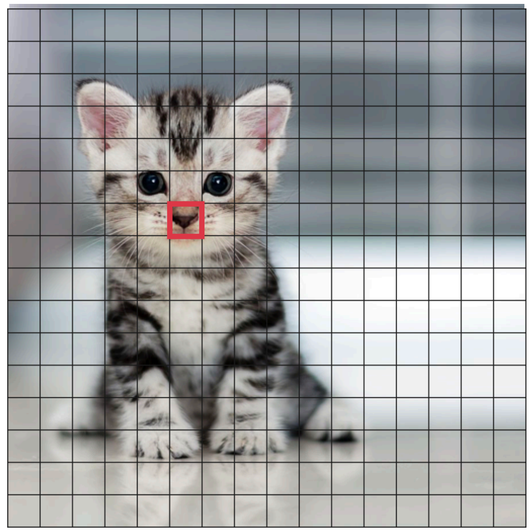
Learning from Self-supervision



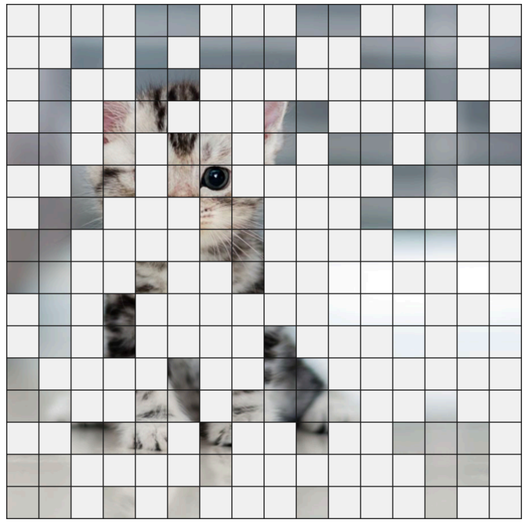
f_θ

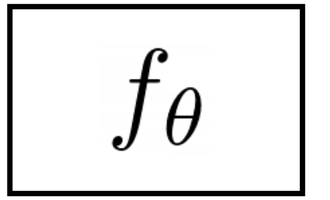
$p(\mathbf{v})$

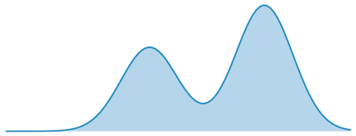
$\mathbf{x} = [0.3, 0.4]$



Learning from Self-supervision

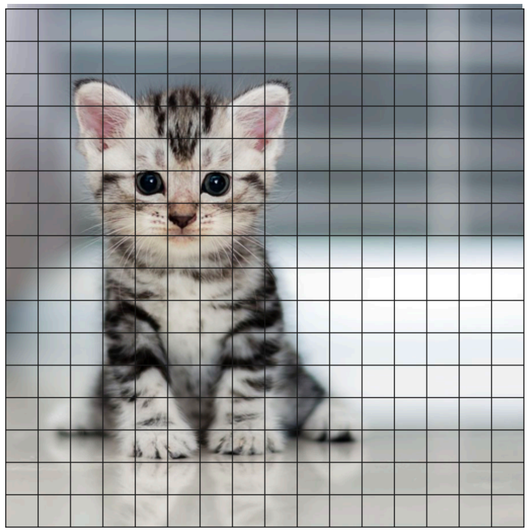






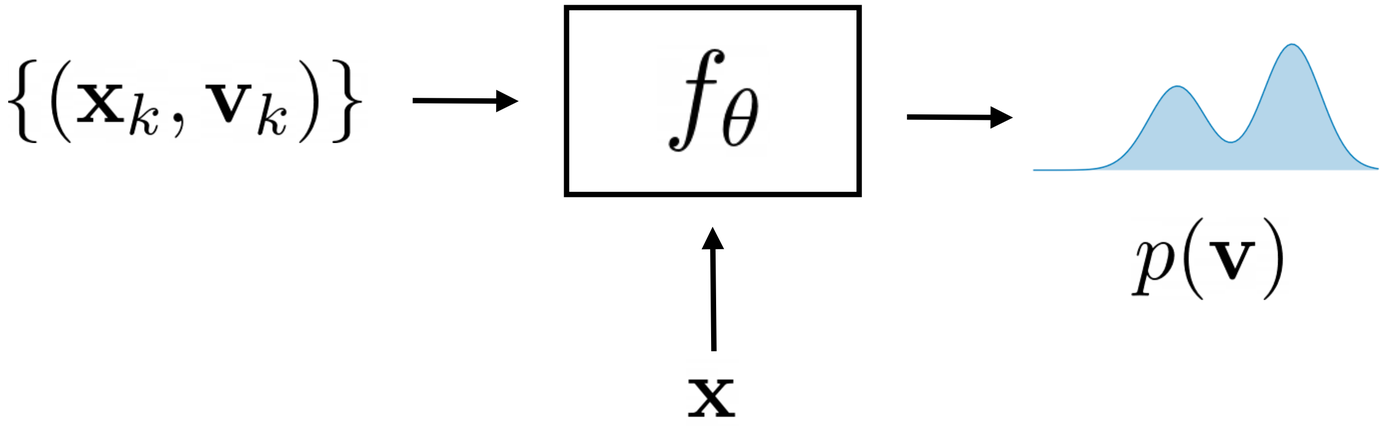
$p(\mathbf{v})$

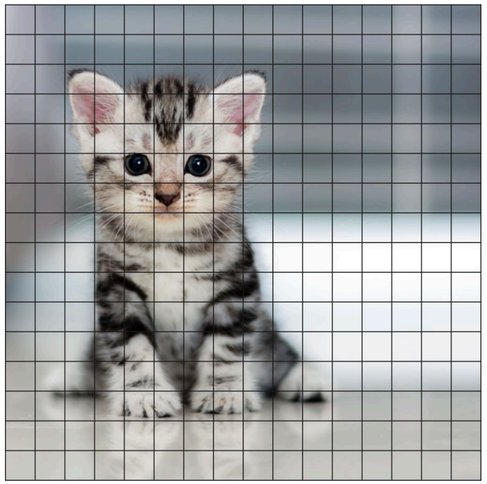
$\mathbf{x} = [0.3, 0.4]$



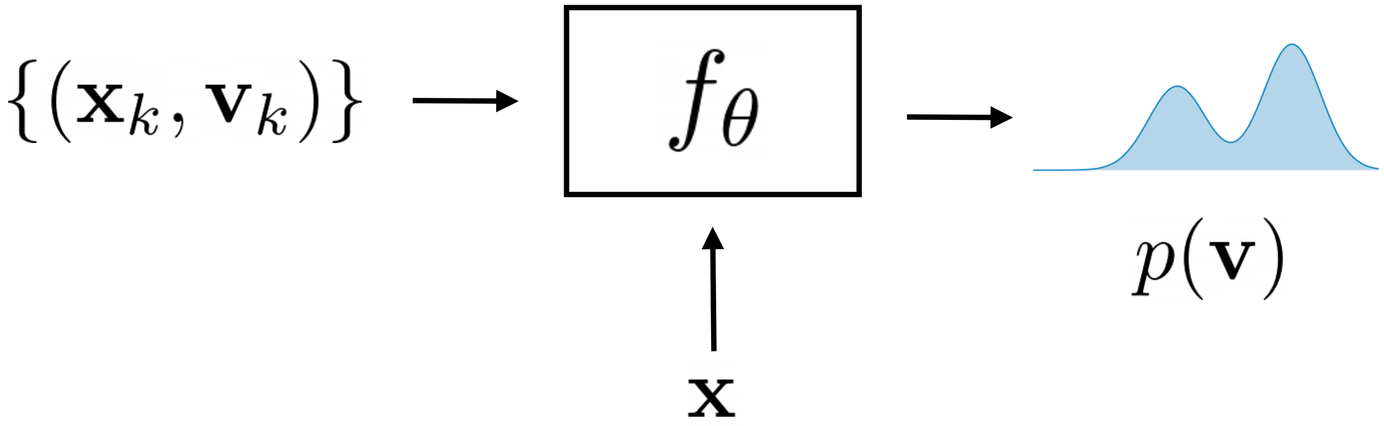
Training Objective : $-\log p($  $;$  $)$

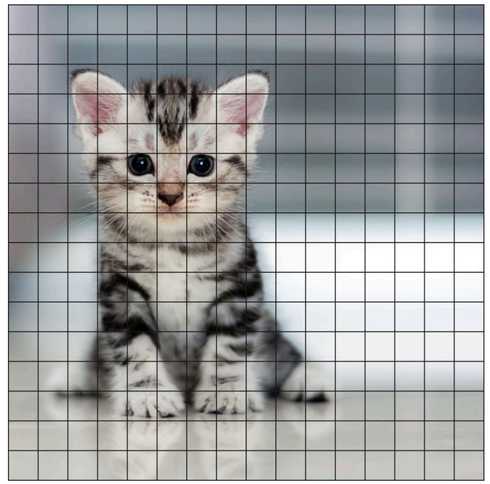
Modeling Generic Spatial Signals





Modeling Generic Spatial Signals





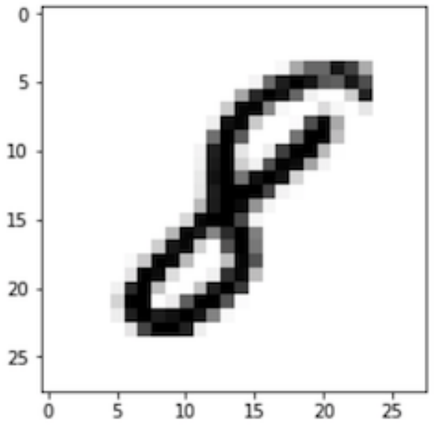
$\mathbf{v} \in \mathbb{R}^3$

$\mathbf{x} \in \mathbb{R}^2$

Modeling Generic Spatial Signals

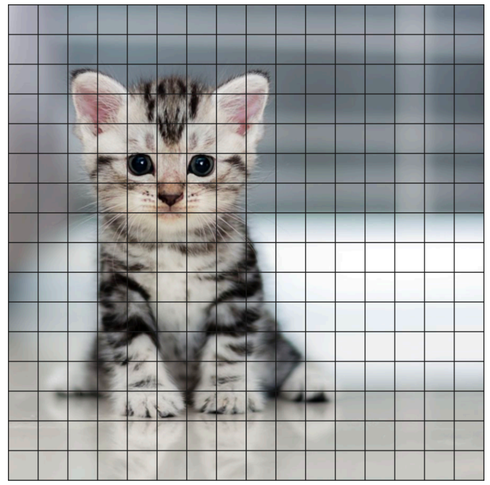
$$\{(\mathbf{x}_k, \mathbf{v}_k)\} \longrightarrow \boxed{f_\theta} \longrightarrow p(\mathbf{v})$$

$$\mathbf{x}$$



$$\mathbf{v} \in \mathbb{R}^1$$

$$\mathbf{x} \in \mathbb{R}^2$$



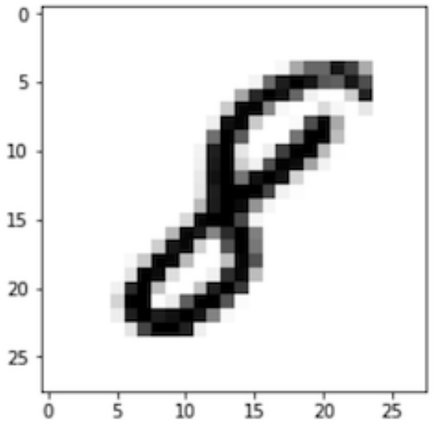
$$\mathbf{v} \in \mathbb{R}^3$$

$$\mathbf{x} \in \mathbb{R}^2$$

Modeling Generic Spatial Signals

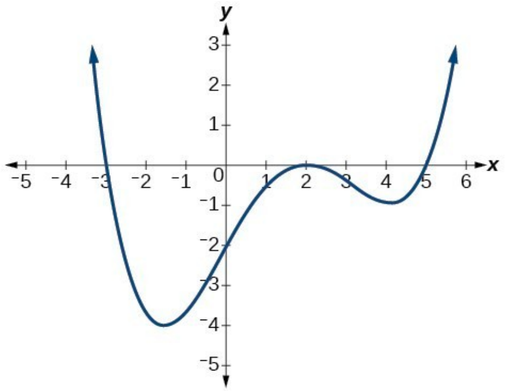
$$\{(\mathbf{x}_k, \mathbf{v}_k)\} \longrightarrow \boxed{f_\theta} \longrightarrow p(\mathbf{v})$$

$$\mathbf{x}$$



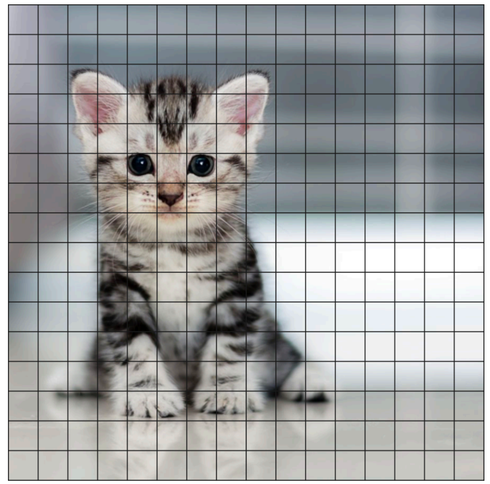
$\mathbf{v} \in \mathbb{R}^1$

$\mathbf{x} \in \mathbb{R}^2$



$\mathbf{v} \in \mathbb{R}^1$

$\mathbf{x} \in \mathbb{R}^1$



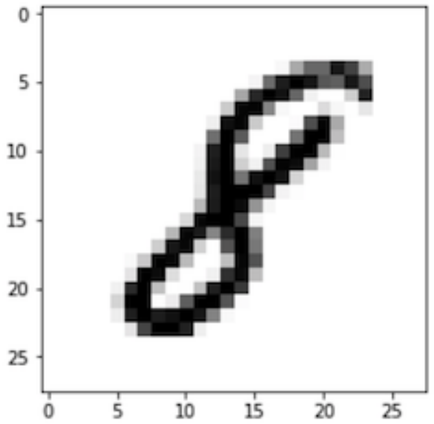
$\mathbf{v} \in \mathbb{R}^3$

$\mathbf{x} \in \mathbb{R}^2$

Modeling Generic Spatial Signals

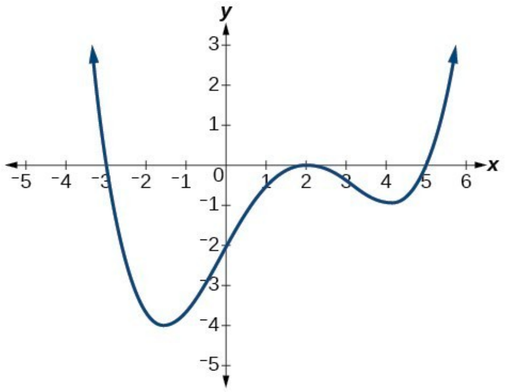
$$\{(\mathbf{x}_k, \mathbf{v}_k)\} \longrightarrow \boxed{f_\theta} \longrightarrow p(\mathbf{v})$$

$$\mathbf{x}$$



$\mathbf{v} \in \mathbb{R}^1$

$\mathbf{x} \in \mathbb{R}^2$

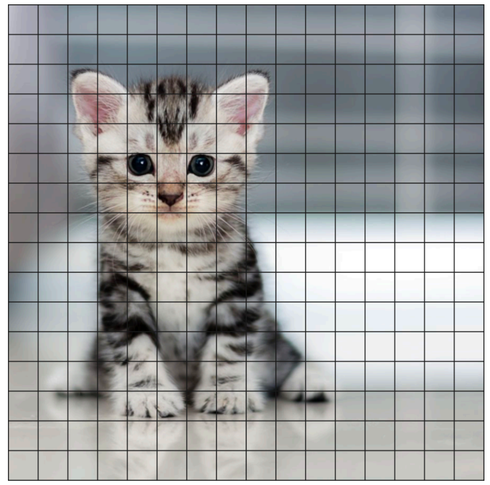


$\mathbf{v} \in \mathbb{R}^1$

$\mathbf{x} \in \mathbb{R}^1$

$\mathbf{v} \in \mathbb{R}^1$

$\mathbf{x} \in \mathbb{R}^3$



$\mathbf{v} \in \mathbb{R}^3$

$\mathbf{x} \in \mathbb{R}^2$

Image Synthesis

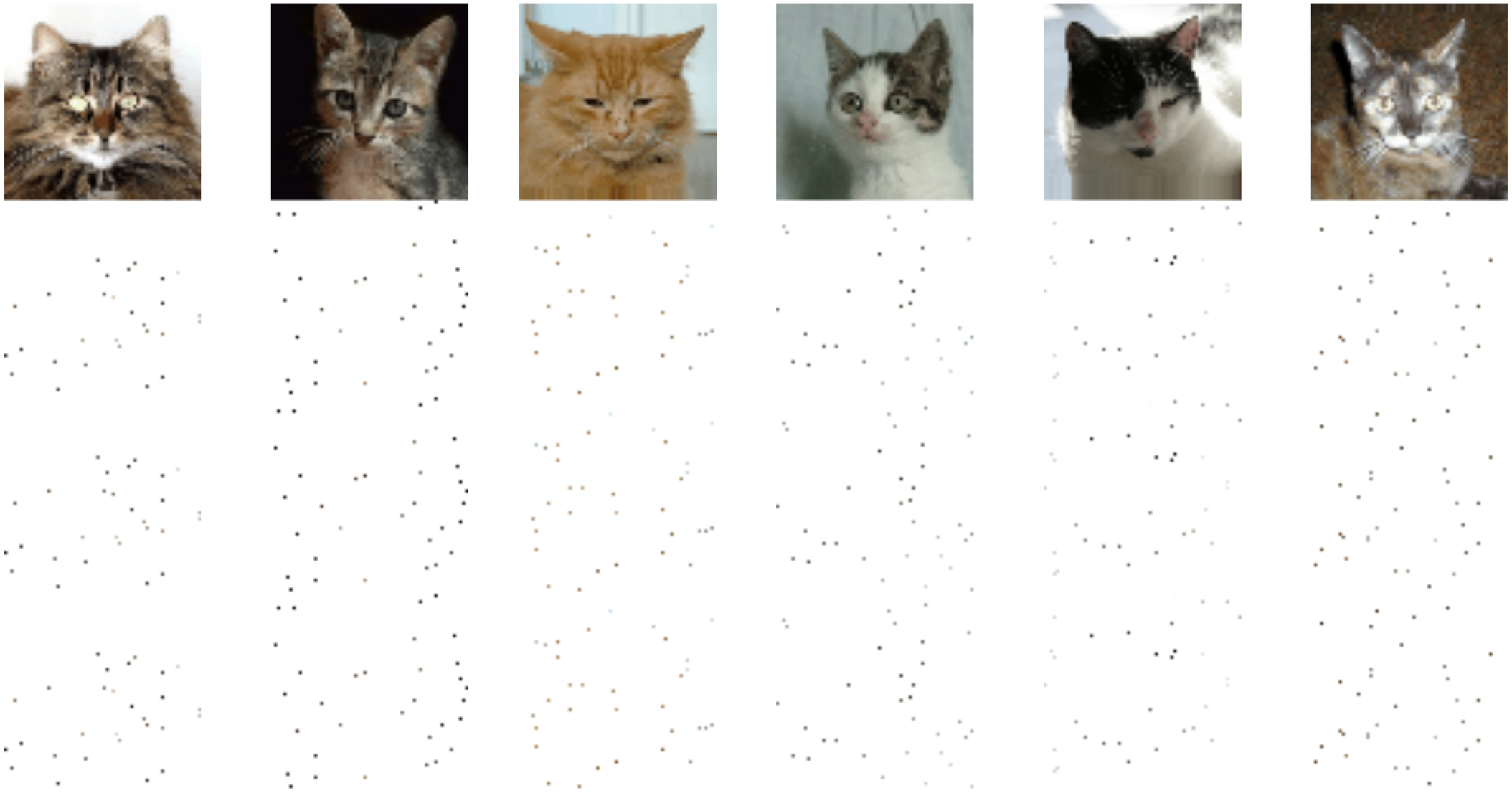


Image Synthesis

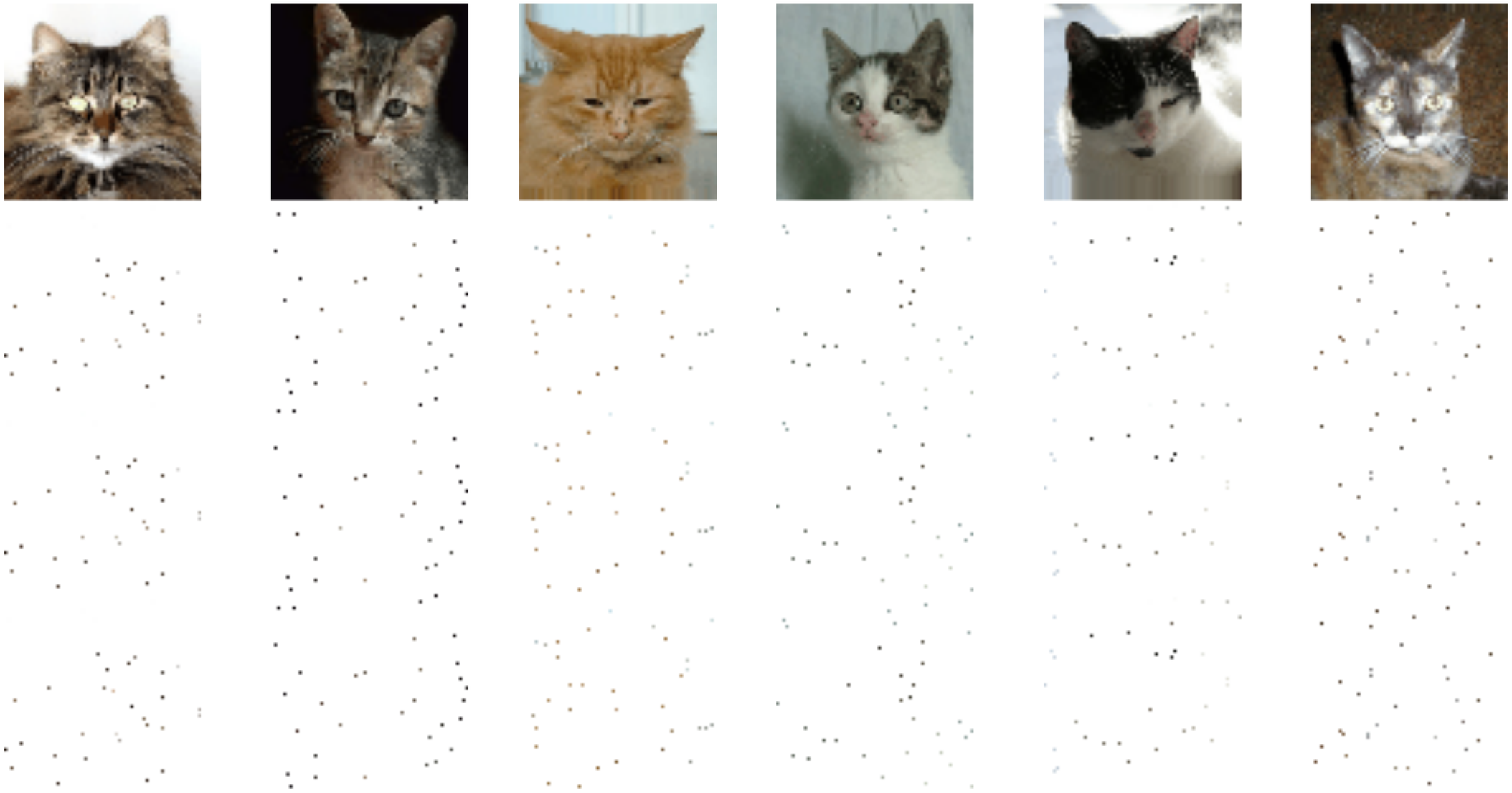
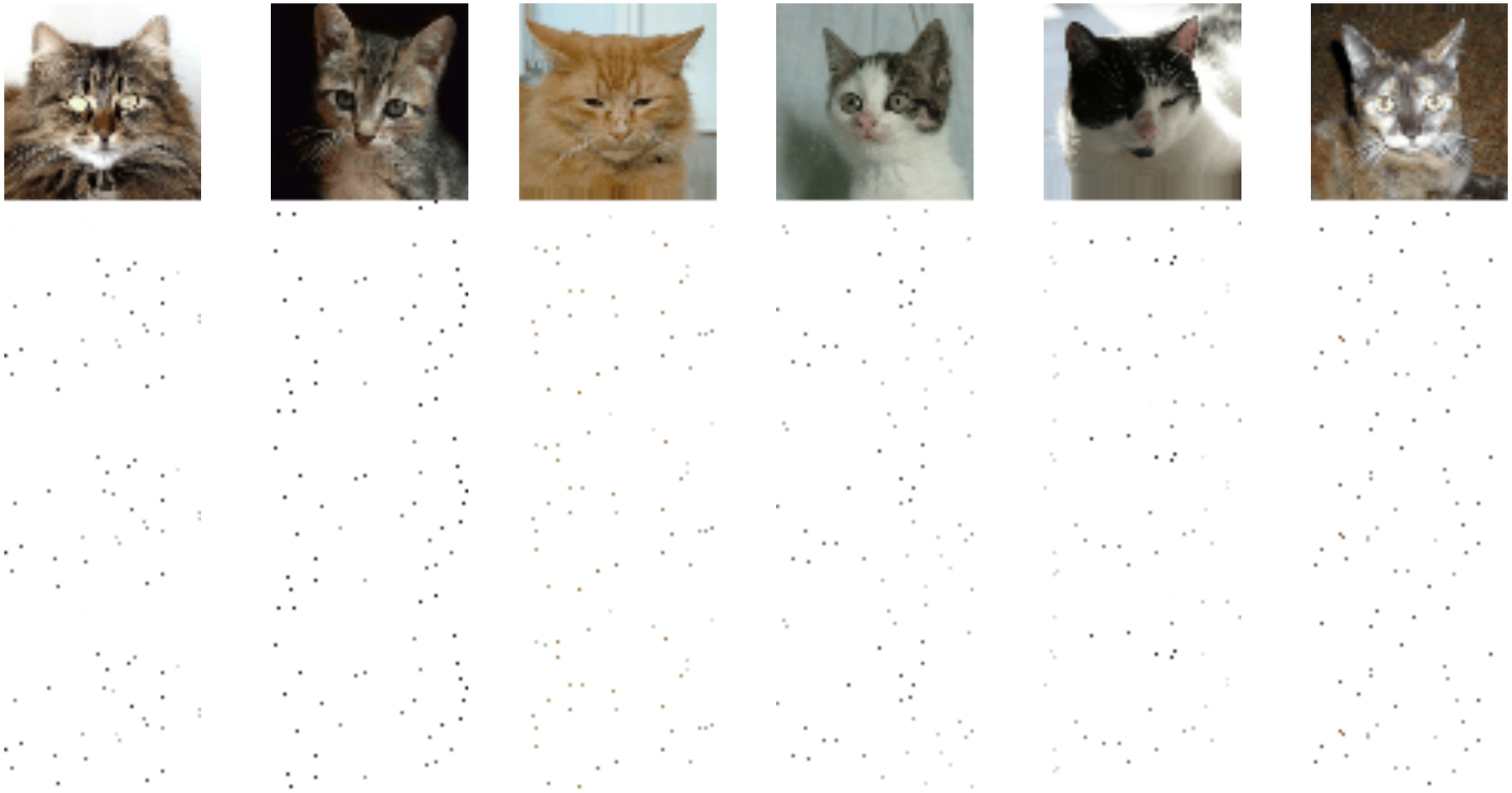
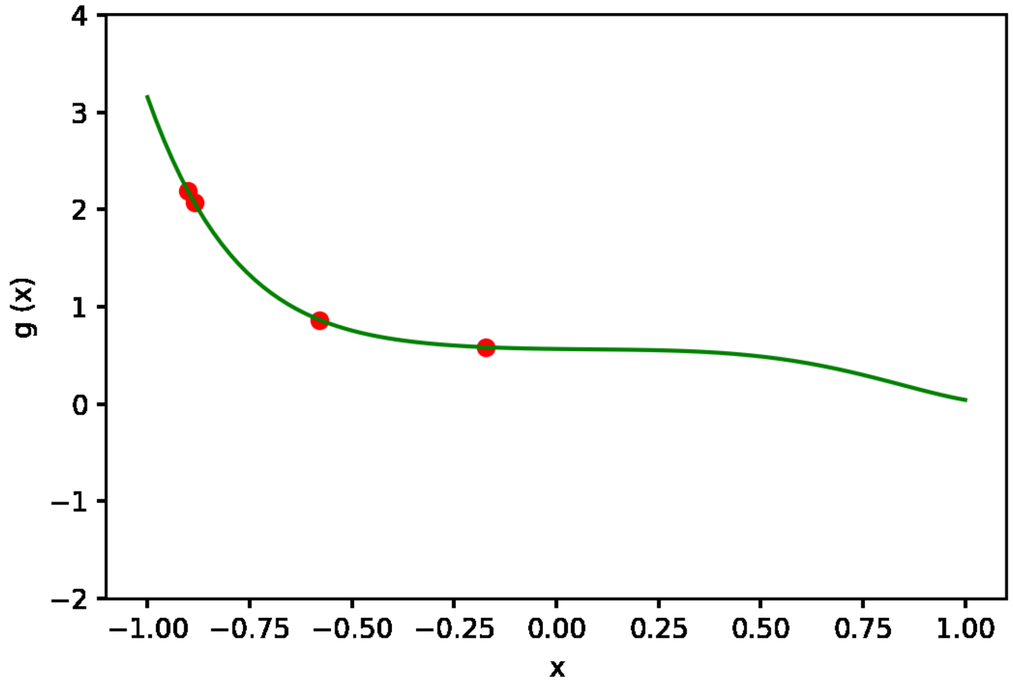
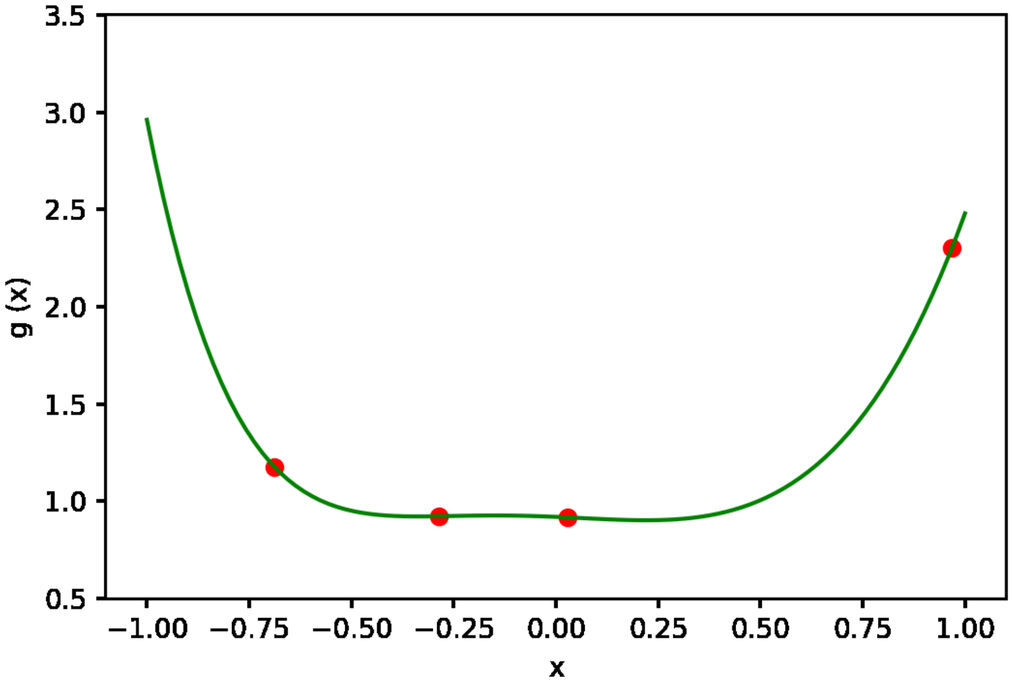


Image Synthesis



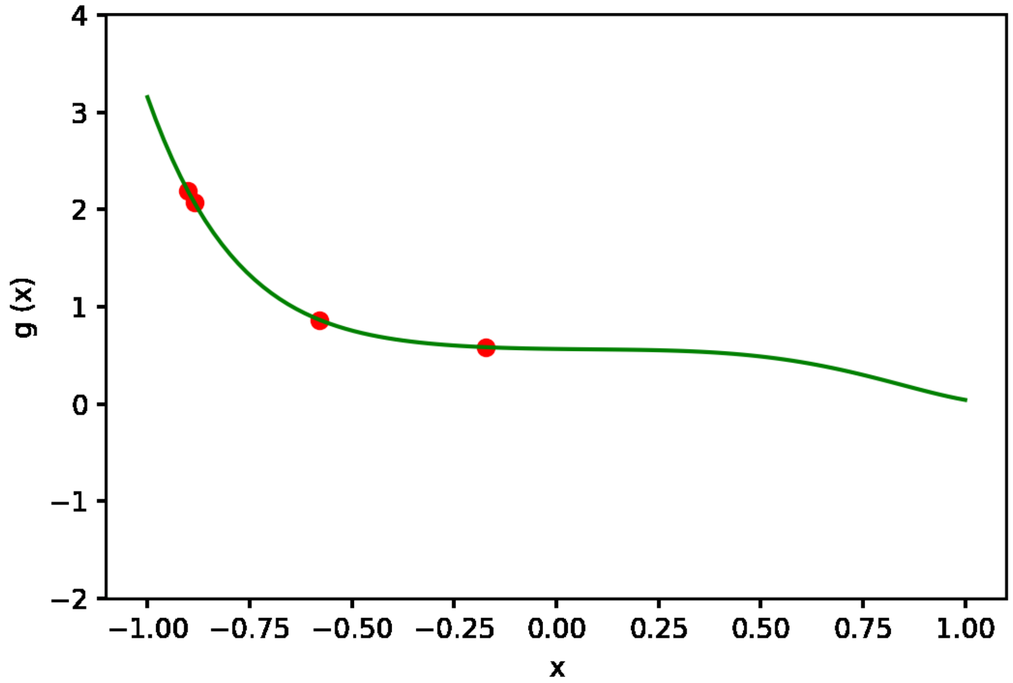
Polynomial Prediction

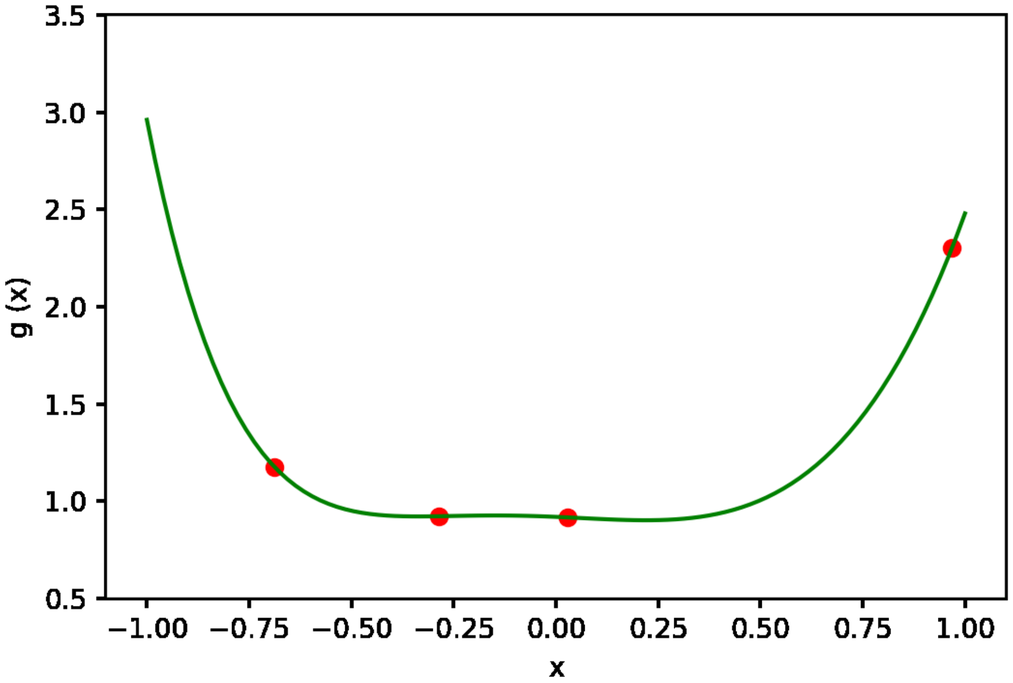




Training data: random degree-6 polynomials

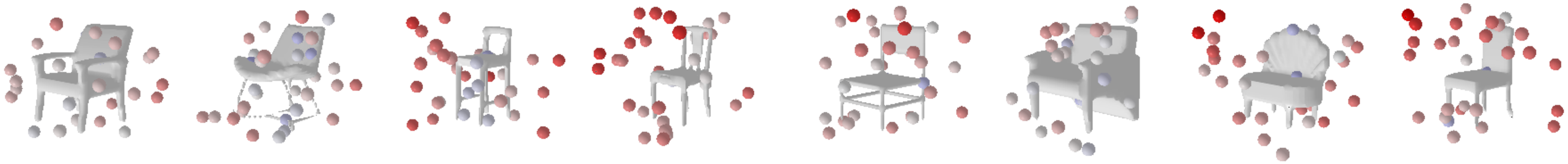
Polynomial Prediction



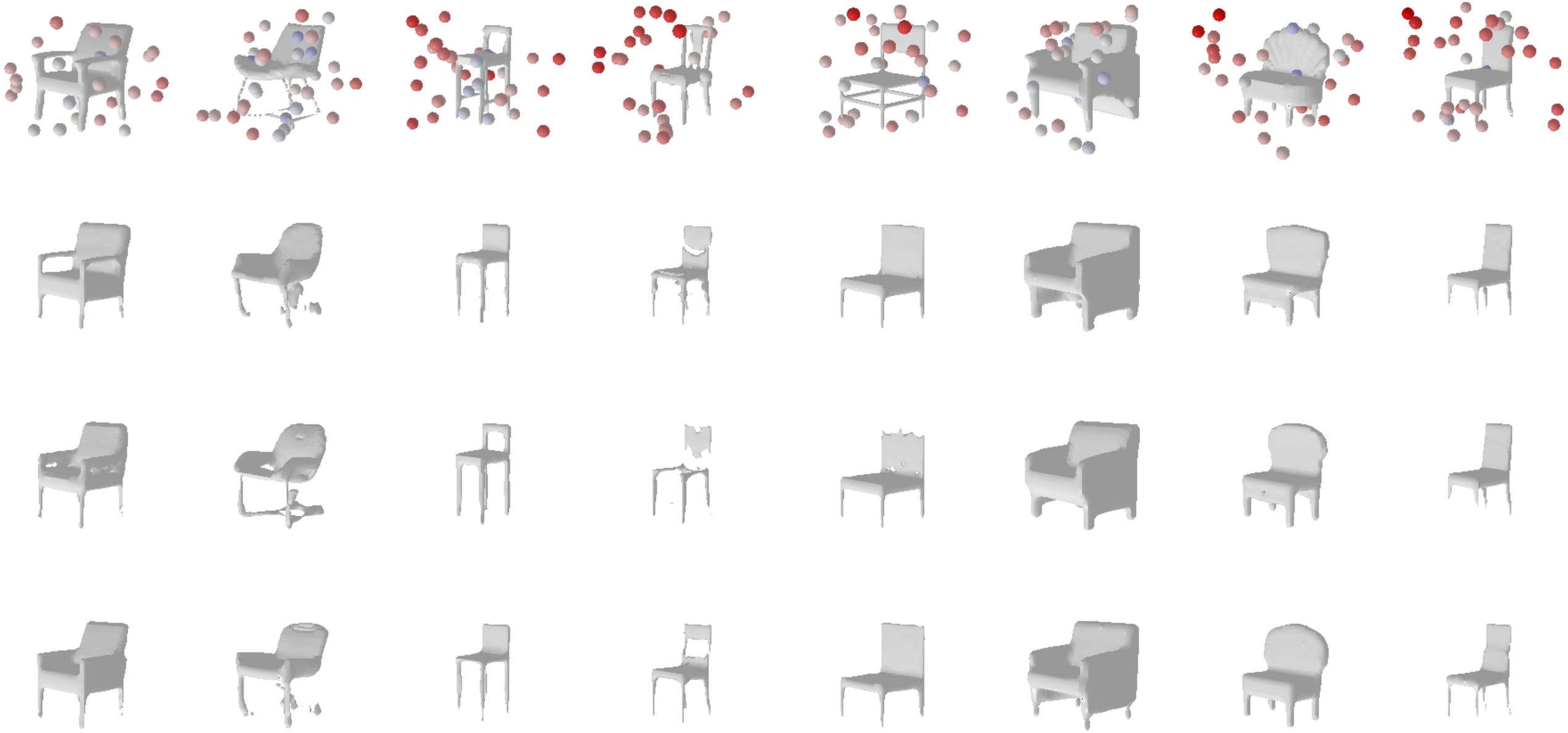


Training data: random degree-6 polynomials

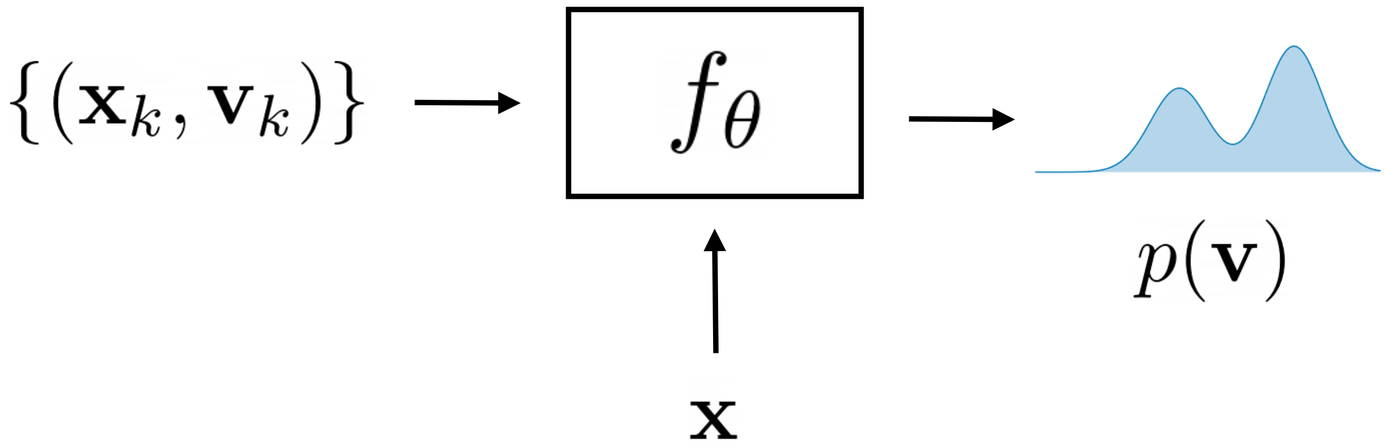
Shape Completion



Shape Completion



Thank you!



Project page: https://shubhtuls.github.io/PixelTransformer/