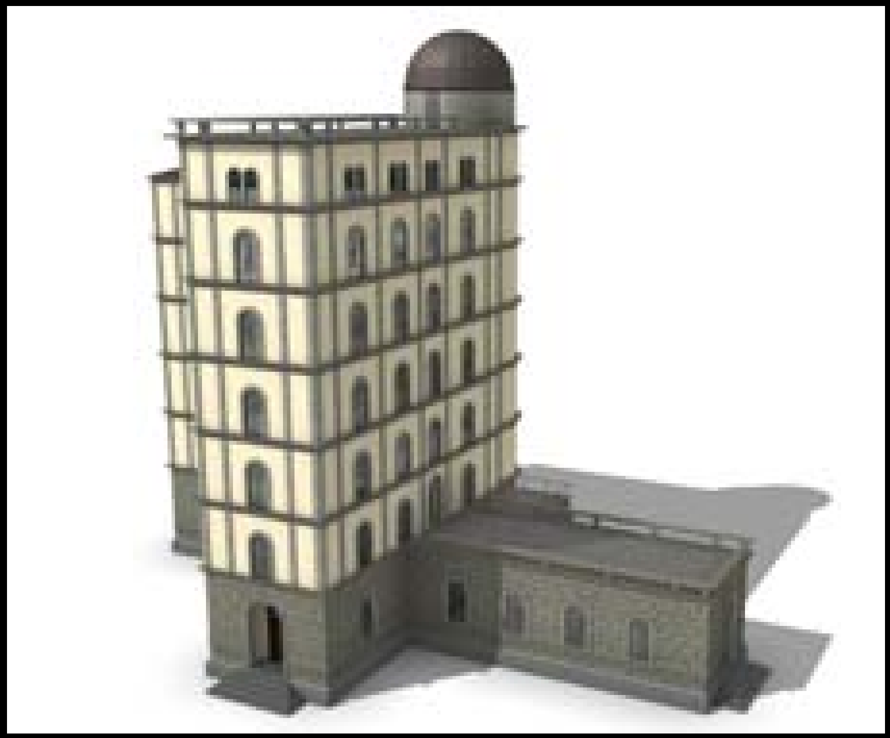
Metropolis Procedural Modeling

Jerry Talton   Yu Lou   Steve Lesser
Jared Duke   Radomír Měch   Vladlen Koltun

Stanford University & Adobe Systems

Grammar-based Models

Grammar-based Models


CFG Formulation

CFG Formulation

Alphabet:

set of *terminal symbols* T and *nonterminal symbols* V

CFG Formulation

Alphabet:

set of *terminal symbols* T and *nonterminal symbols* V

Axiom:

initial string ω

CFG Formulation

Alphabet:

set of *terminal symbols* T and *nonterminal symbols* V

Axiom:

initial string ω

Rewriting rules:

predecessor nonterminal → *successor* symbols

CFG Formulation

Alphabet:

set of *terminal symbols* T and *nonterminal symbols* V

Axiom:

initial string ω

Rewriting rules:

predecessor nonterminal → *successor* symbols

Turtle interpretation:

F draw line - turn left + turn right [push] pop

CFG Example

CFG Example

$$\omega : X$$
$$X \rightarrow F[-X][+X]$$

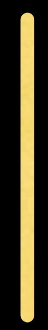
CFG Example

$$\omega : X$$
$$X \rightarrow F[-X][+X]$$

X

CFG Example

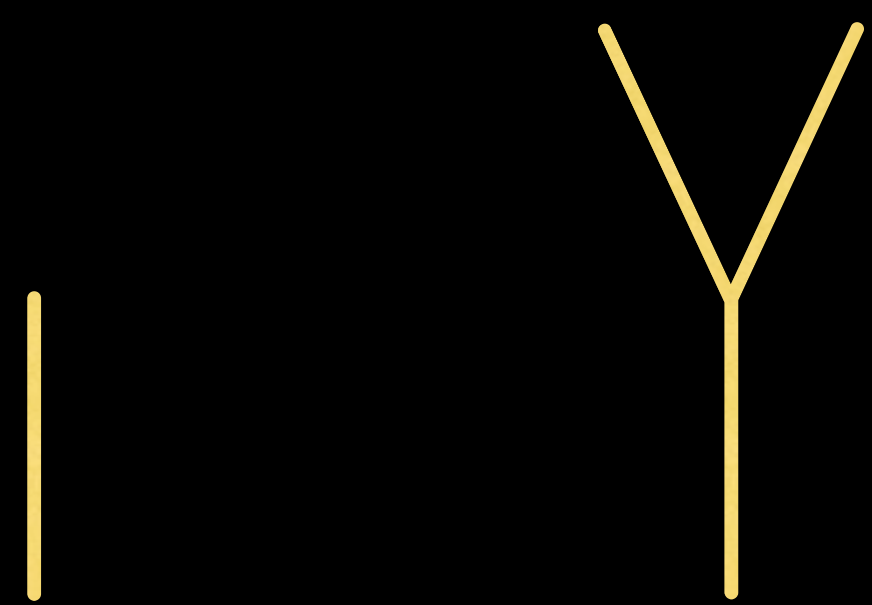
$$\omega : X$$
$$X \rightarrow F[-X][+X]$$



X $\quad$ $F[-X][+X]$

CFG Example

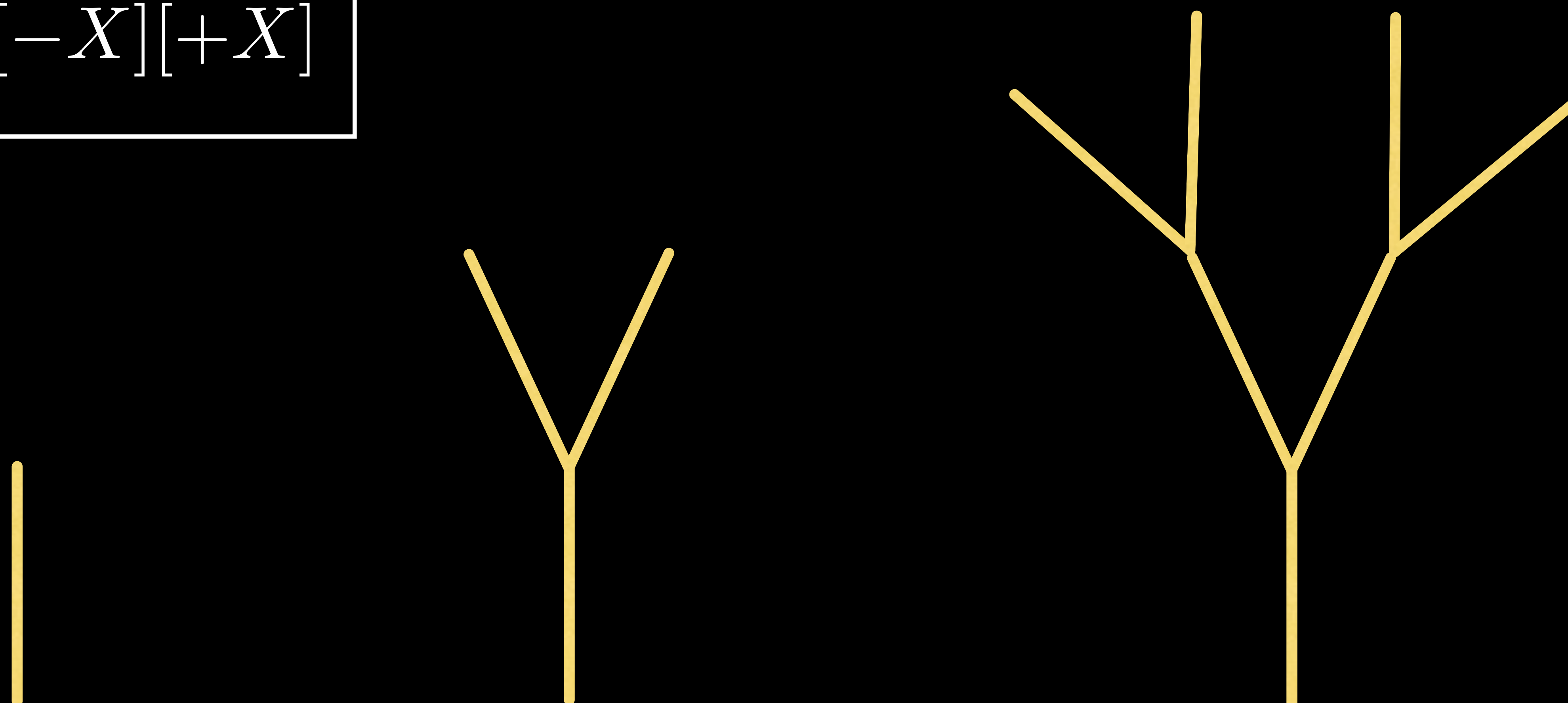
$$\omega : X$$
$$X \rightarrow F[-X][+X]$$



X $\quad F[-X][+X]$ $\quad F[-F[-X][+X]][+F[-X][+X]]$

CFG Example

$$\omega : X$$
$$X \rightarrow F[-X][+X]$$



X $F[-X][+X]$ $F[-F[-X][+X]][+F[-X][+X]]$ $\ldots$

Stochastic CFGs

Stochastic CFGs

- Assign probability to each rewriting rule in grammar G

Stochastic CFGs

- Assign probability to each rewriting rule in grammar G
- Yields distribution $\pi(\cdot)$ over space of derivations $\mathcal{L}(G)$

Stochastic CFGs

- Assign probability to each rewriting rule in grammar G
- Yields distribution $\pi(\cdot)$ over space of derivations $\mathcal{L}(G)$
- Gives *generative model* that can be **sampled**

Stochastic CFG Example

Stochastic CFG Example

$$\omega : X$$

$$X \stackrel{.5}{\rightarrow} F[-X][+X]$$

$$X \stackrel{.25}{\rightarrow} F[+X]$$

$$X \stackrel{.25}{\rightarrow} F[-X]$$

Stochastic CFG Example

$$\omega : X$$

$$X \stackrel{.5}{\rightarrow} F[-X][+X]$$

$$X \stackrel{.25}{\rightarrow} F[+X]$$

$$X \stackrel{.25}{\rightarrow} F[-X]$$

1

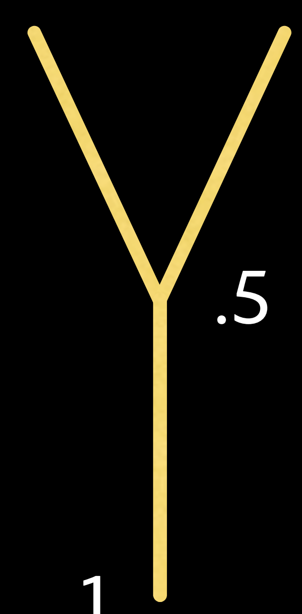
Stochastic CFG Example

$$\omega : X$$

$$X \overset{.5}{\rightarrow} F[-X][+X]$$

$$X \overset{.25}{\rightarrow} F[+X]$$

$$X \overset{.25}{\rightarrow} F[-X]$$



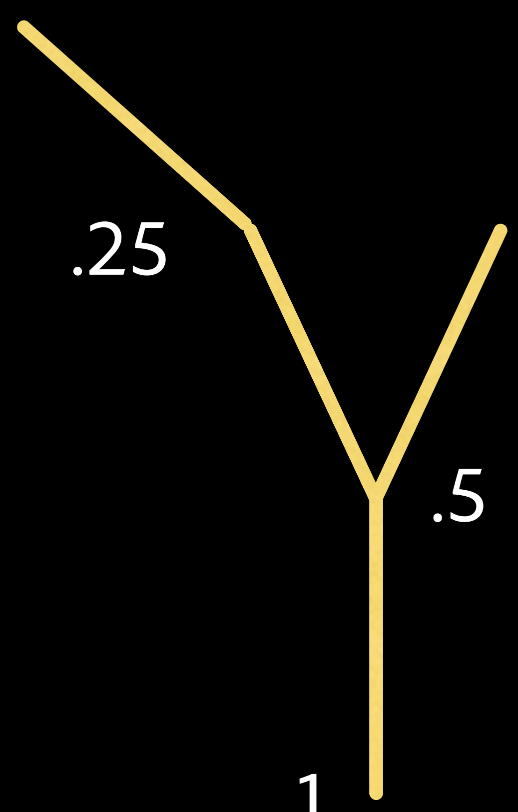
Stochastic CFG Example

$$\omega : X$$

$$X \xrightarrow{.5} F[-X][+X]$$

$$X \xrightarrow{.25} F[+X]$$

$$X \xrightarrow{.25} F[-X]$$



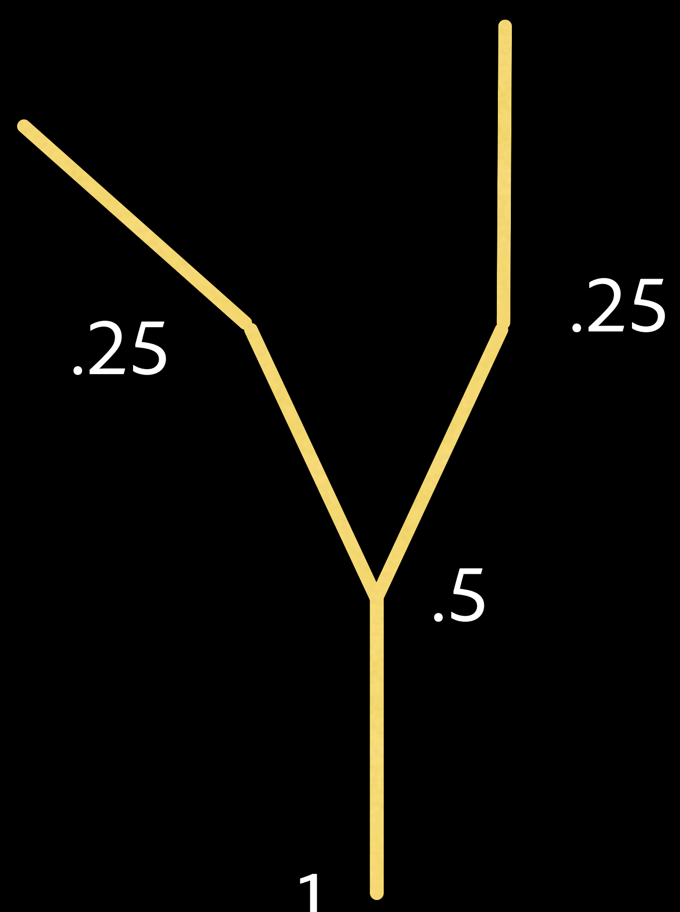
Stochastic CFG Example

$$\omega : X$$

$$X \xrightarrow{.5} F[-X][+X]$$

$$X \xrightarrow{.25} F[+X]$$

$$X \xrightarrow{.25} F[-X]$$



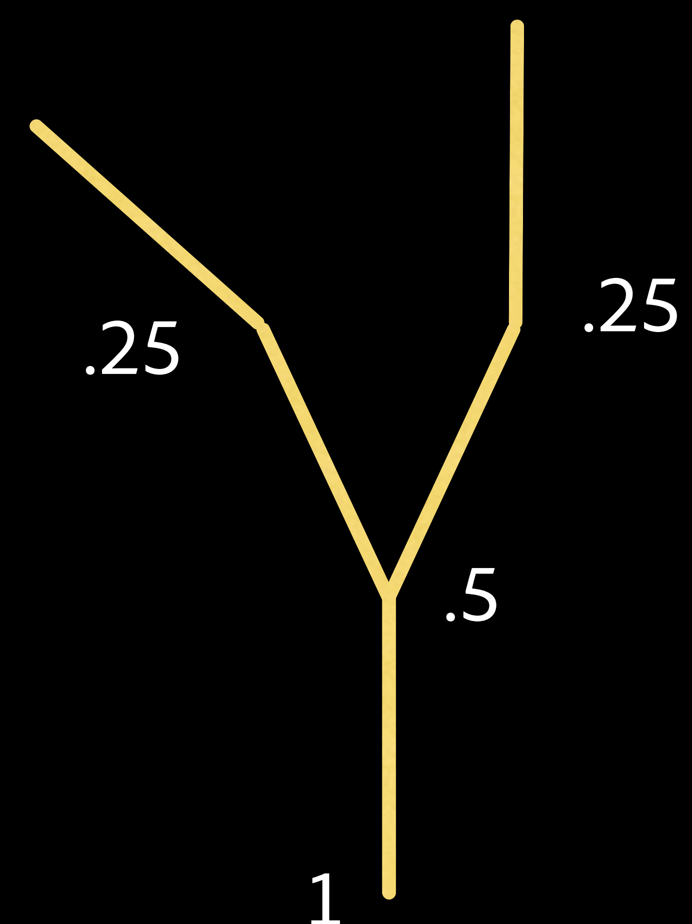
Stochastic CFG Example

$$\omega : X$$

$$X \stackrel{.5}{\rightarrow} F[-X][+X]$$

$$X \stackrel{.25}{\rightarrow} F[+X]$$

$$X \stackrel{.25}{\rightarrow} F[-X]$$



$$\pi(\delta) = 1 \times .5 \times .25 \times .25$$

Parametric CFGs

Parametric CFGs

$\omega : X$

Parametric CFGs

$$\omega : X$$

$$X \rightarrow F(t)[-X][+X]$$

- Terminal symbols associated with numeric parameters

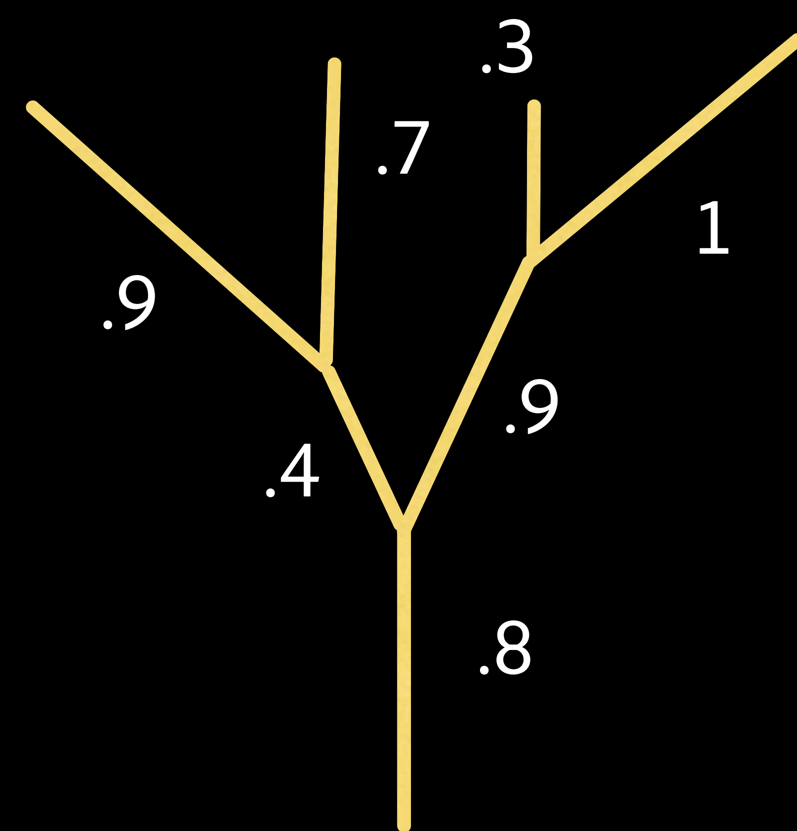
Parametric CFGs

$$\omega : X$$
$$X \to F(t)[-X][+X]$$
$$t \sim N(\mu, \sigma^2)$$

- Terminal symbols associated with numeric parameters
- Each parameter ϕ_i sampled from a distribution $\Phi_i(\cdot)$

Parametric CFGs

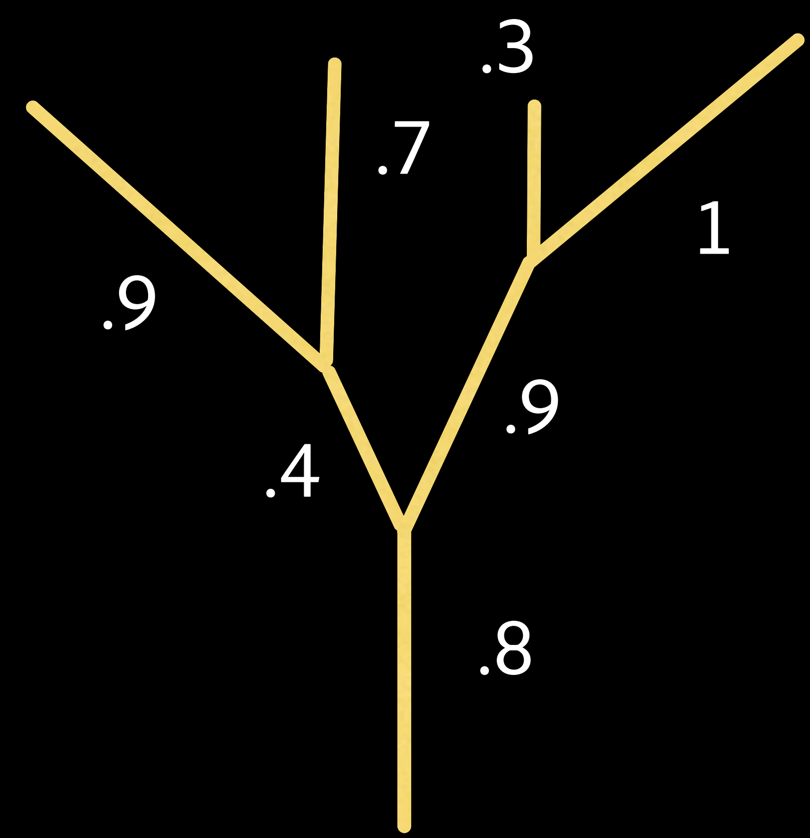
$$\omega : X$$
$$X \rightarrow F(t)[-X][+X]$$
$$t \sim N(\mu, \sigma^2)$$



- Terminal symbols associated with numeric parameters
- Each parameter ϕ_i sampled from a distribution $\Phi_i(\cdot)$
- Parameters control visual appearance of components

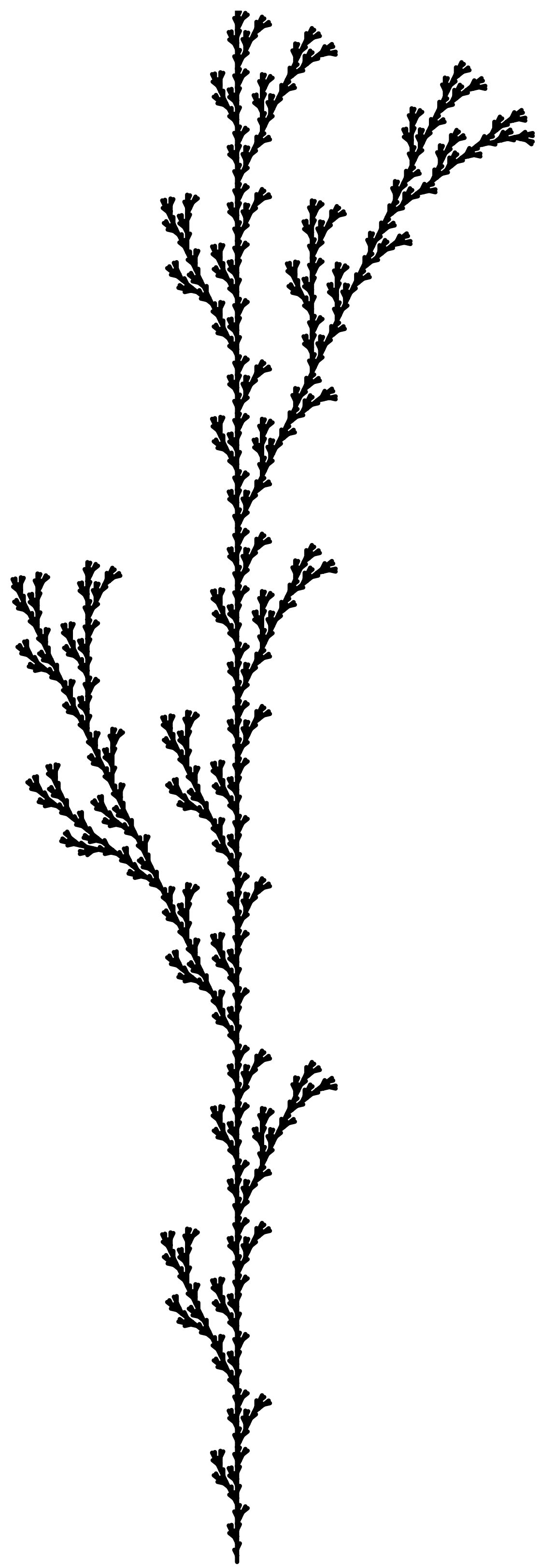
Parametric CFGs

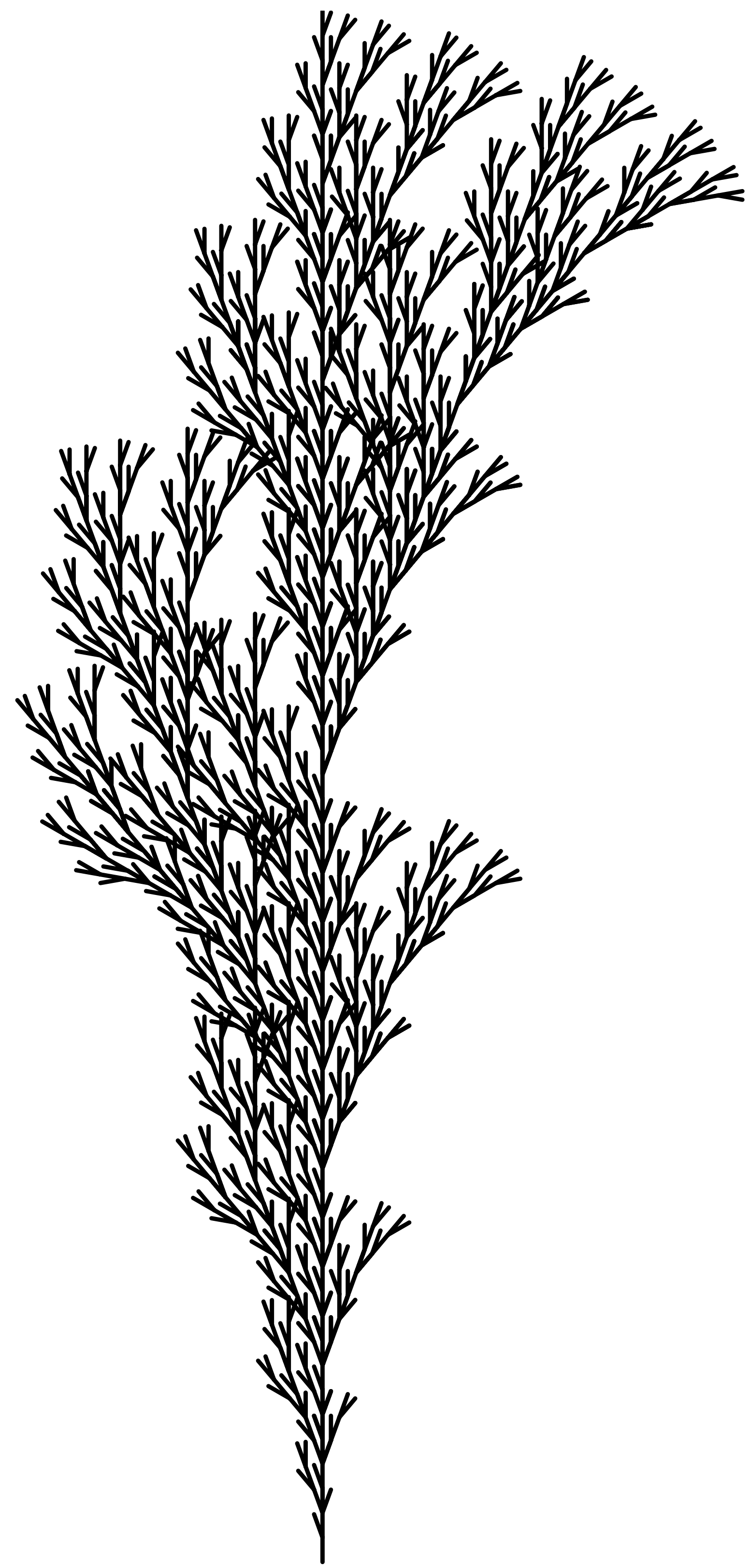
$$\omega : X$$
$$X \rightarrow F(t)[-X][+X]$$
$$t \sim N(\mu, \sigma^2)$$



$\phi =$ [.8, .4, .9, .9, .7, .3, 1]

- Terminal symbols associated with numeric parameters
- Each parameter ϕ_i sampled from a distribution $\Phi_i(\cdot)$
- Parameters control visual appearance of components

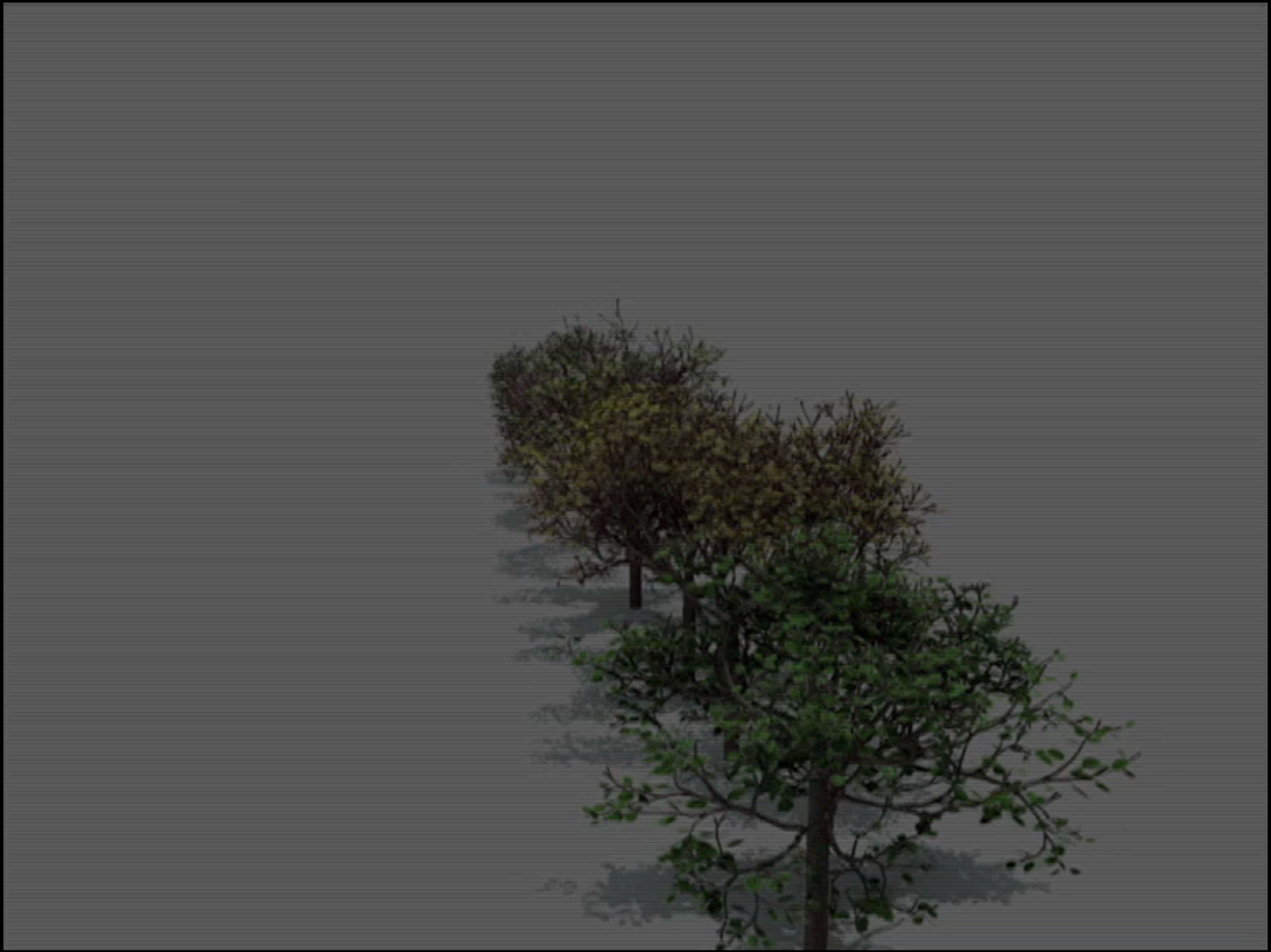




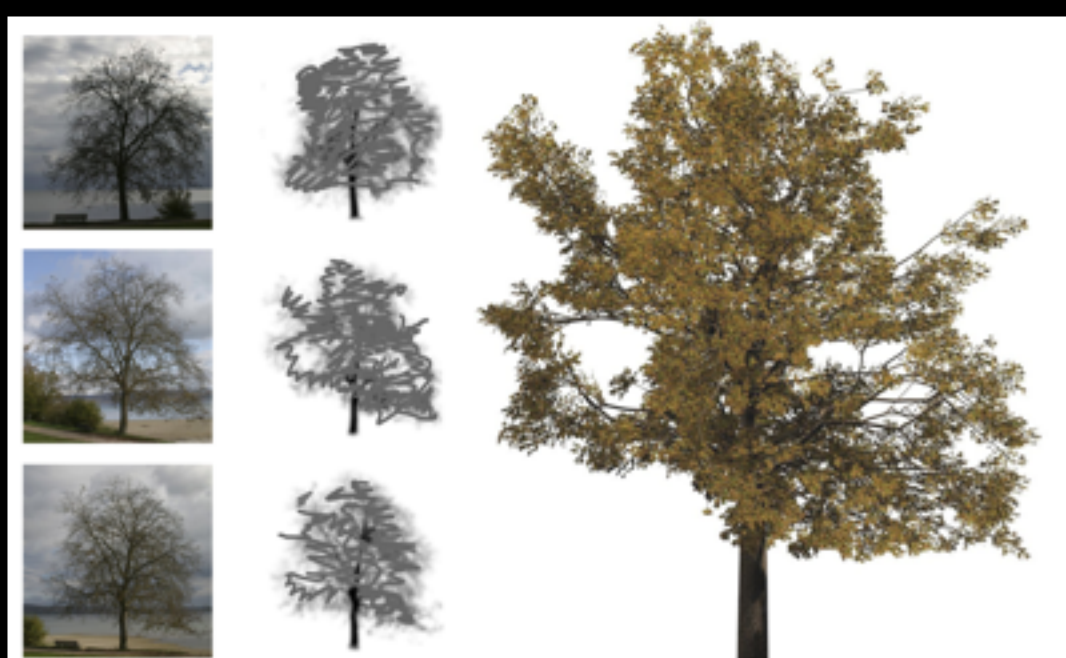
$$F \to F[+F]F[-F]F \qquad\qquad F \to F[+F]F[-F][F]$$

From "The Algorithmic Beauty of Plants"

Our contribution: general method for bringing artistic *control* to grammar-based models

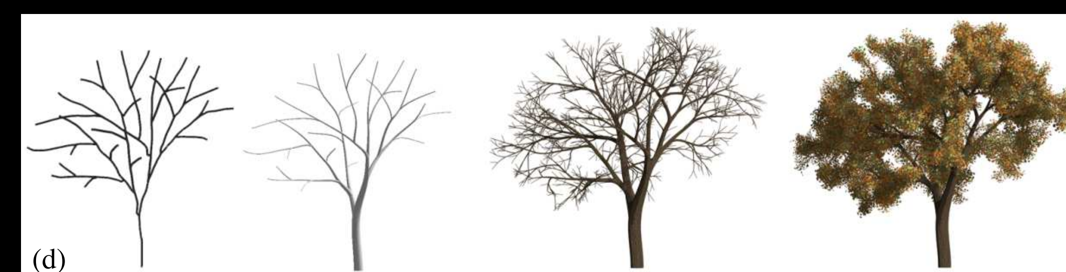


[Prusinkiewicz *et al.* ’94]



[Neubert *et al.* ’07]

[Aliaga *et al.* ’07]



[Chen *et al.* ’08]

[Teboul *et al.* ’10]

Goal: decouple model specification from control mechanism

Modeling as Optimization

Modeling as Optimization

Bayesian Inference

Modeling as Optimization

Bayesian Inference

- consider space of derivations $\delta \in \Delta(G)$

Modeling as Optimization

Bayesian Inference

- consider space of derivations $\delta \in \Delta(G)$
- define model prior $\pi(\delta)$

Modeling as Optimization

Bayesian Inference

- consider space of derivations $\delta \in \Delta(G)$
- define model prior $\pi(\delta)$
- take in some user specification I

Modeling as Optimization

Bayesian Inference

- consider space of derivations $\delta \in \Delta(G)$
- define model prior $\pi(\delta)$
- take in some user specification I
- formulate likelihood function $L(I|\delta)$

Modeling as Optimization

Bayesian Inference

- consider space of derivations $\delta \in \Delta(G)$
- define model prior $\pi(\delta)$
- take in some user specification I
- formulate likelihood function $L(I|\delta)$
- maximize $p(\delta|I) \propto L(I|\delta)\pi(\delta)$

Modeling as Optimization

Bayesian Inference

- consider space of derivations $\delta \in \Delta(G)$
- define model prior $\pi(\delta)$
- take in some user specification I
- formulate likelihood function $L(I|\delta)$
- maximize $p(\delta|I) \propto L(I|\delta)\pi(\delta)$

the tricky bit

Key idea: simulate a *Markov Chain* to sample from $p(\cdot|\cdot)$

Key idea: simulate a *Markov Chain* to sample from $p(\cdot|\cdot)$, perform *maximum a posteriori* estimation

MCMC Review

A *Markov Chain* is a sequence of random variables $X_1, X_2, \ldots$ with the *Markov Property*:

$$P(X_n = x | X_{n-1} = x_{n-1}, \ldots, X_1 = x_1) = P(X_n = x | X_{n-1} = x_{n-1})$$

MCMC Review

A *Markov Chain* is a sequence of random variables $X_1, X_2, \ldots$ with the *Markov Property*:

$$P(X_n = x | X_{n-1} = x_{n-1}, \ldots, X_1 = x_1) = P(X_n = x | X_{n-1} = x_{n-1})$$

Properly constructed, each $X_i \sim p(X)$, where $p(X)$ is the *stationary distribution* of the chain

Metropolis-Hastings

Metropolis-Hastings

Pick a *proposal density* $q(X_*|X_n)$ that can be sampled from efficiently

Metropolis-Hastings

Pick a *proposal density* $q(X_*|X_n)$ that can be sampled from efficiently

At each step:

Metropolis-Hastings

Pick a *proposal density* $q(X_*|X_n)$ that can be sampled from efficiently

At each step:

- draw $X_* \sim q(\cdot|X_n)$

Metropolis-Hastings

Pick a *proposal density* $q(X_*|X_n)$ that can be sampled from efficiently

At each step:

- draw $X_* \sim q(\cdot|X_n)$
- Compute an *acceptance probability*

$$\alpha = \min\left(\frac{p(X_*)}{p(X_i)}\frac{q(X_n|X_*)}{q(X_*|X_n)}, 1\right)$$

Metropolis-Hastings

Pick a *proposal density* $q(X_*|X_n)$ that can be sampled from efficiently

At each step:

- draw $X_* \sim q(\cdot|X_n)$
- Compute an *acceptance probability*

$$\alpha = \min\left(\frac{p(X_*)}{p(X_i)}\frac{q(X_n|X_*)}{q(X_*|X_n)}, 1\right)$$

- Accept $X_{n+1} = X_*$ or reject $X_{n+1} = X_n$

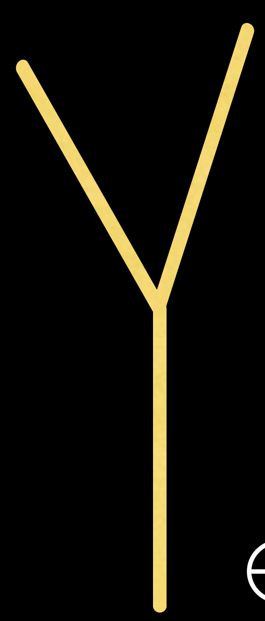
An Opportunity

MH algorithm lets us *sample* efficiently from any function we can *evaluate*...

A Conundrum

MH algorithm lets us *sample* efficiently from any function defined over a fixed-dimensional space we can *evaluate*...

A Conundrum

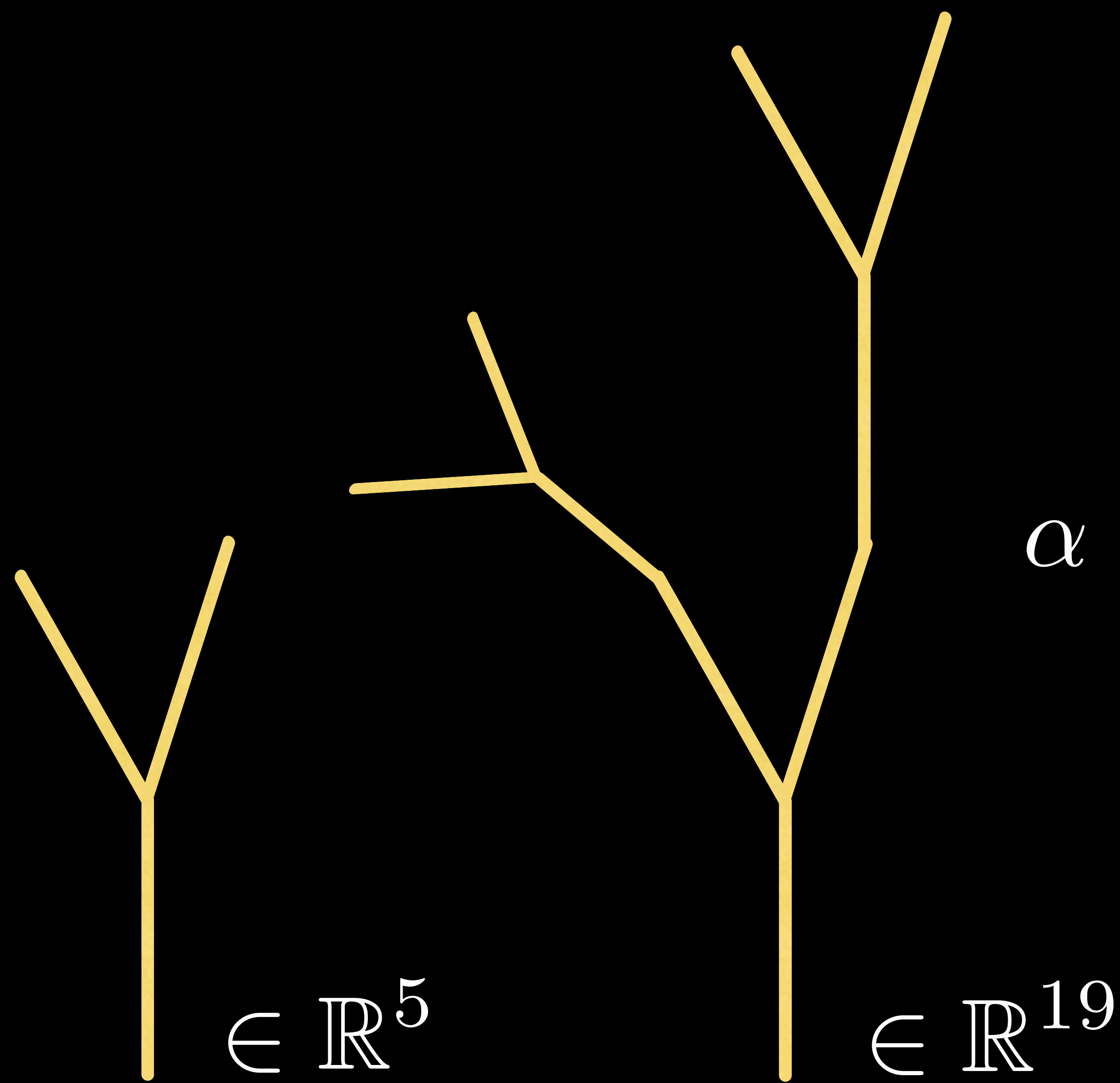


$\in \mathbb{R}^5$

A Conundrum

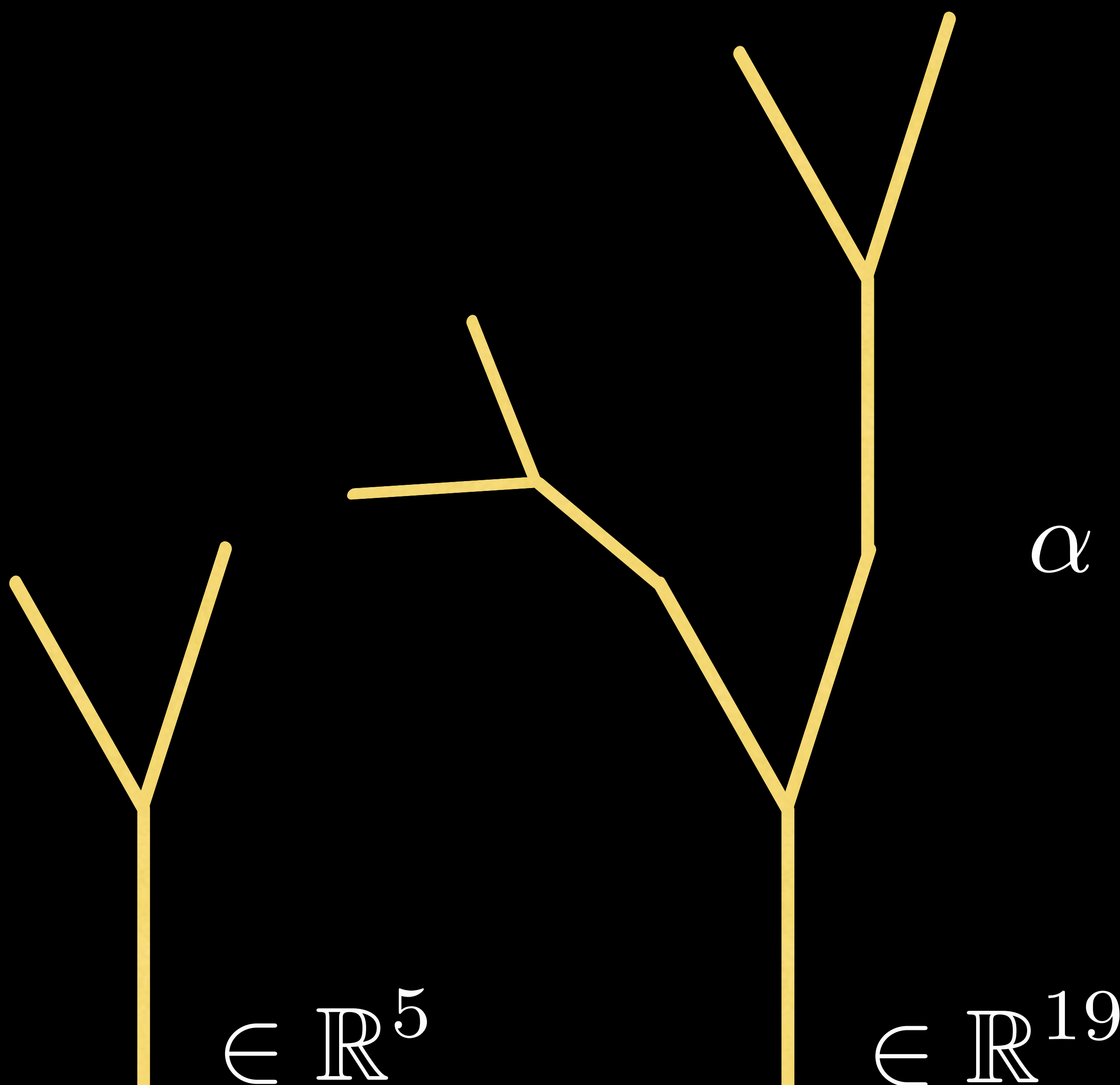
$\in \mathbb{R}^5$ $\in \mathbb{R}^{19}$

A Conundrum



$$\alpha = \min\left(\frac{p(X_*)}{p(X_i)}\frac{q(X_n|X_*)}{q(X_*|X_n)}, 1\right)$$

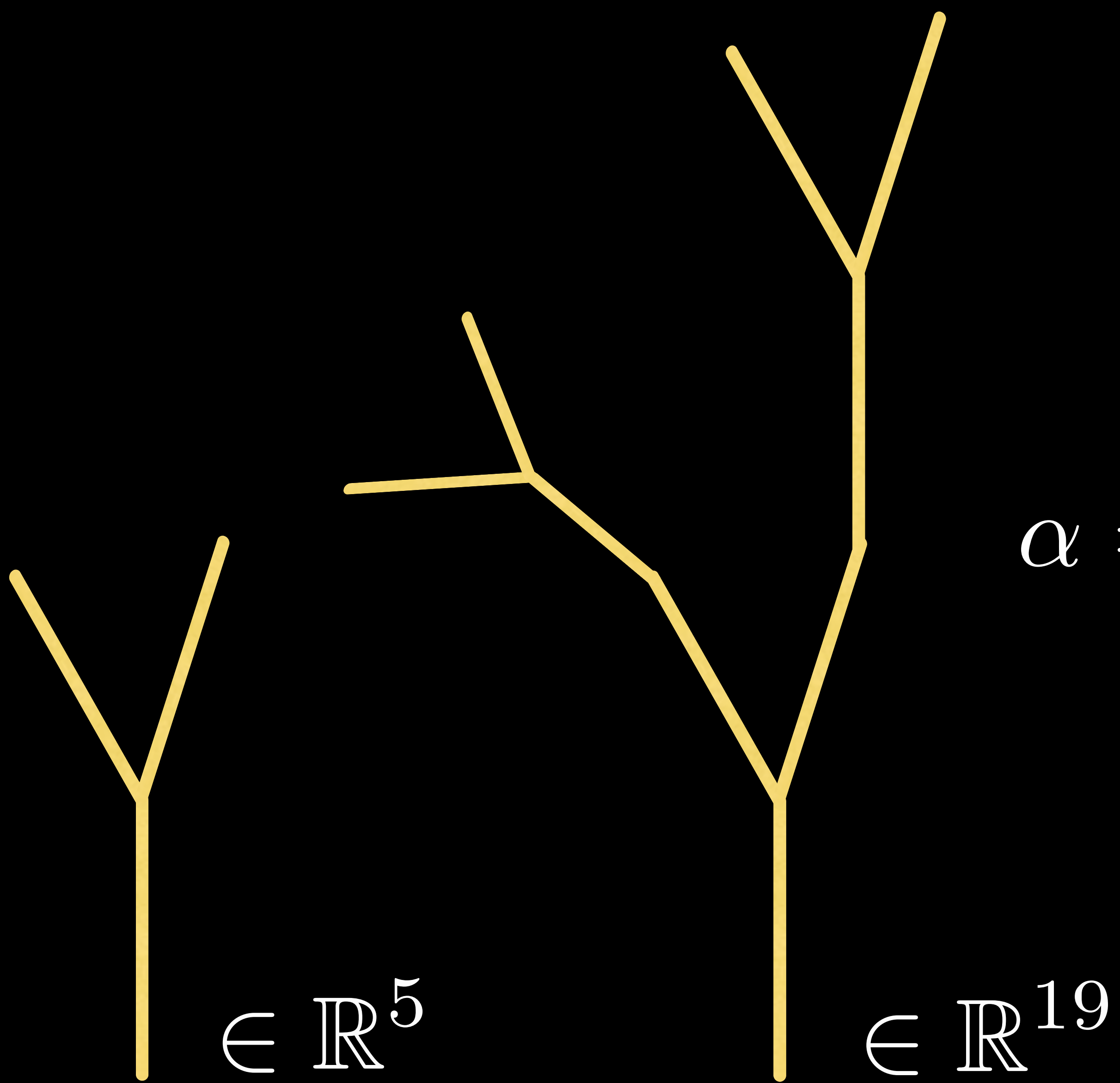
A Conundrum



lives in $\mathbb{R}^{19}$

$$\alpha = \min\left(\frac{p(X_*)}{p(X_i)}\frac{q(X_n|X_*)}{q(X_*|X_n)}, 1\right)$$

A Conundrum



lives in $\mathbb{R}^{19}$

$$\alpha = \min\left(\frac{p(X_*)}{p(X_i)}\frac{q(X_n|X_*)}{q(X_*|X_n)}, 1\right)$$

lives in $\mathbb{R}^5$

Reversible Jump MCMC

Reversible Jump MCMC

- Introduced by [Green '95]

Reversible Jump MCMC

- Introduced by [Green ’95]
- Trans-dimensional MCMC

Reversible Jump MCMC

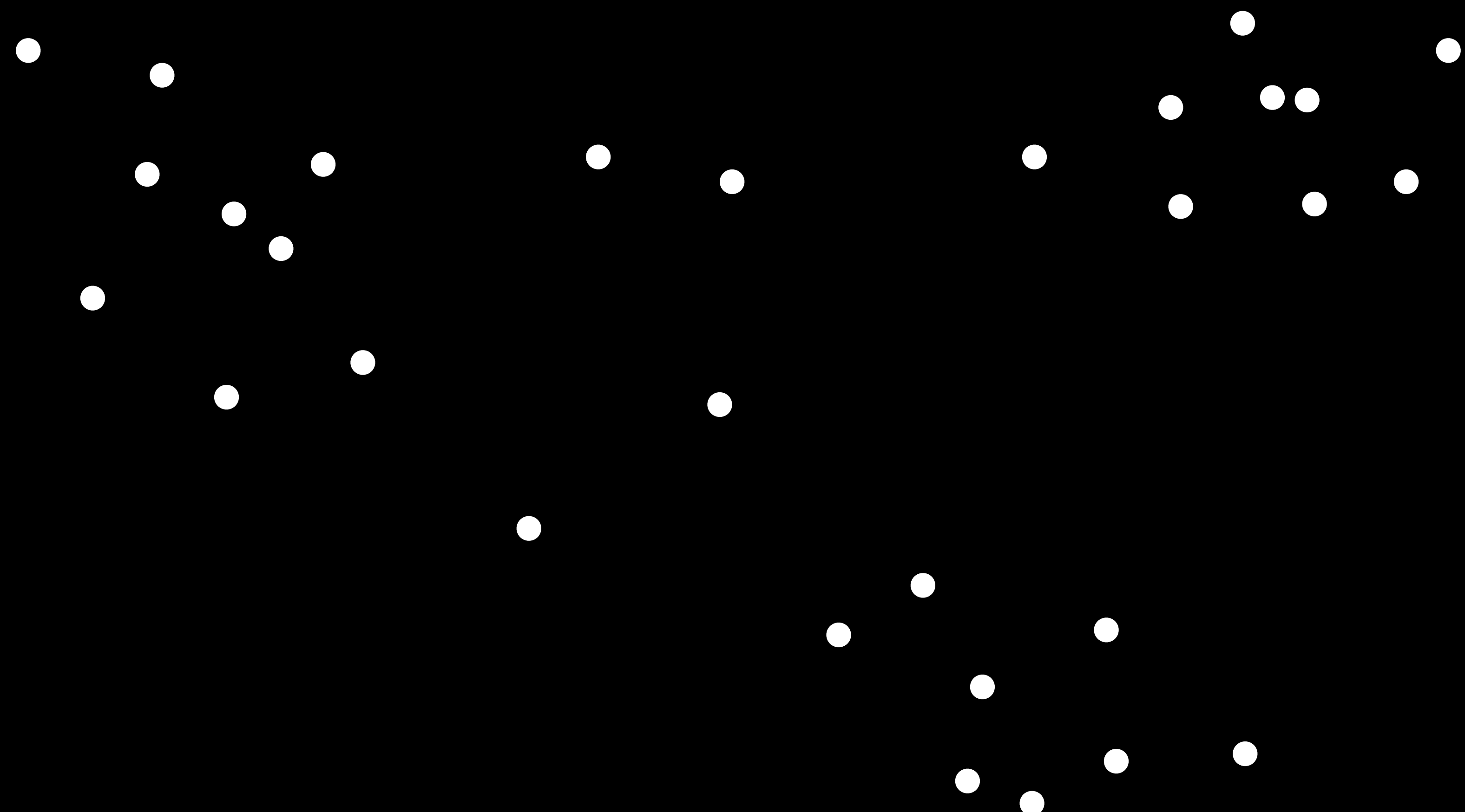
- Introduced by [Green '95]
- Trans-dimensional MCMC
- Extends MH from *parameter fitting* to *model selection*

Reversible Jump MCMC

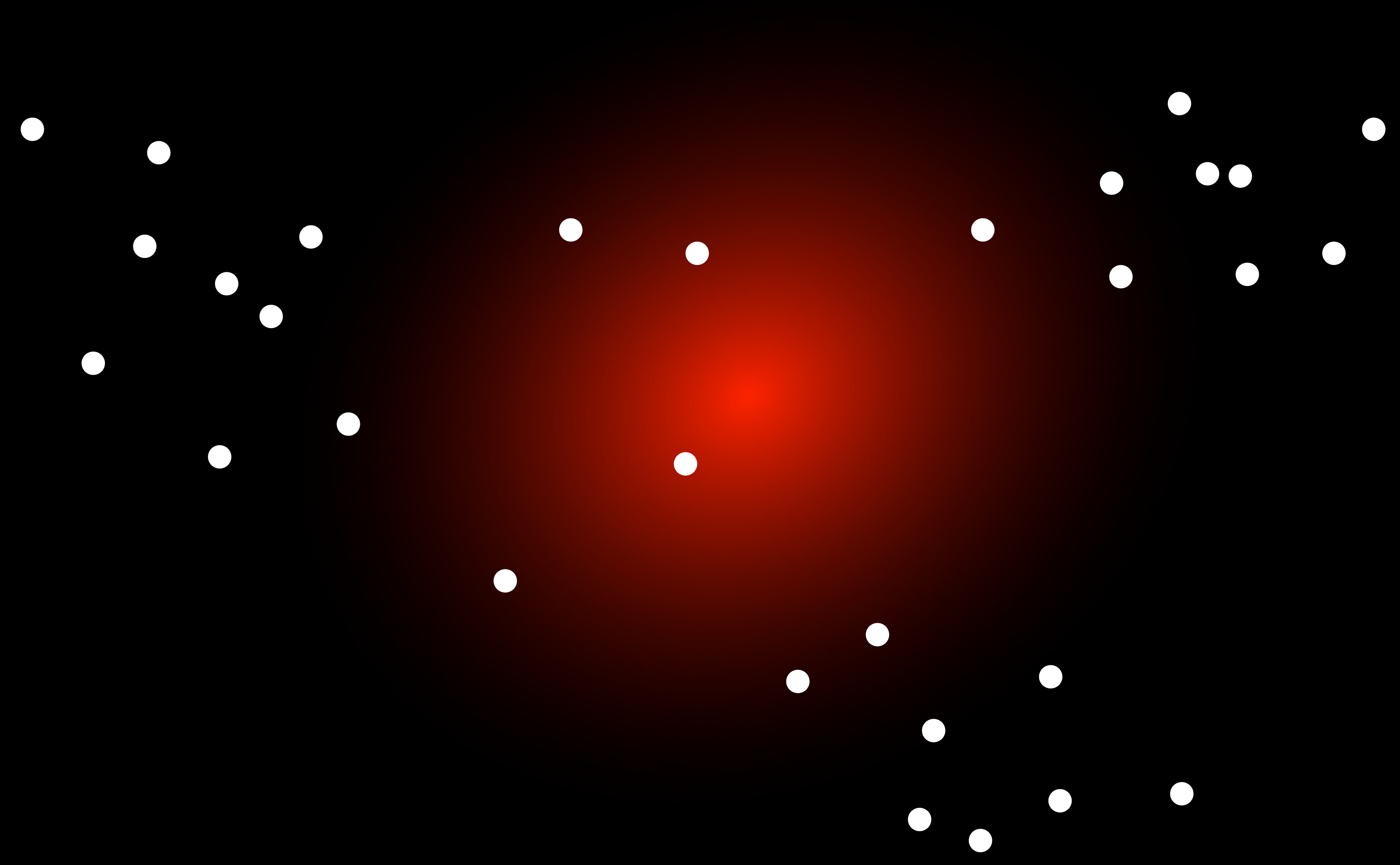
- Introduced by [Green '95]
- Trans-dimensional MCMC
- Extends MH from *parameter fitting* to *model selection*

Use RJMCMC when "*the number of things you don't know is one of the things you don't know*"

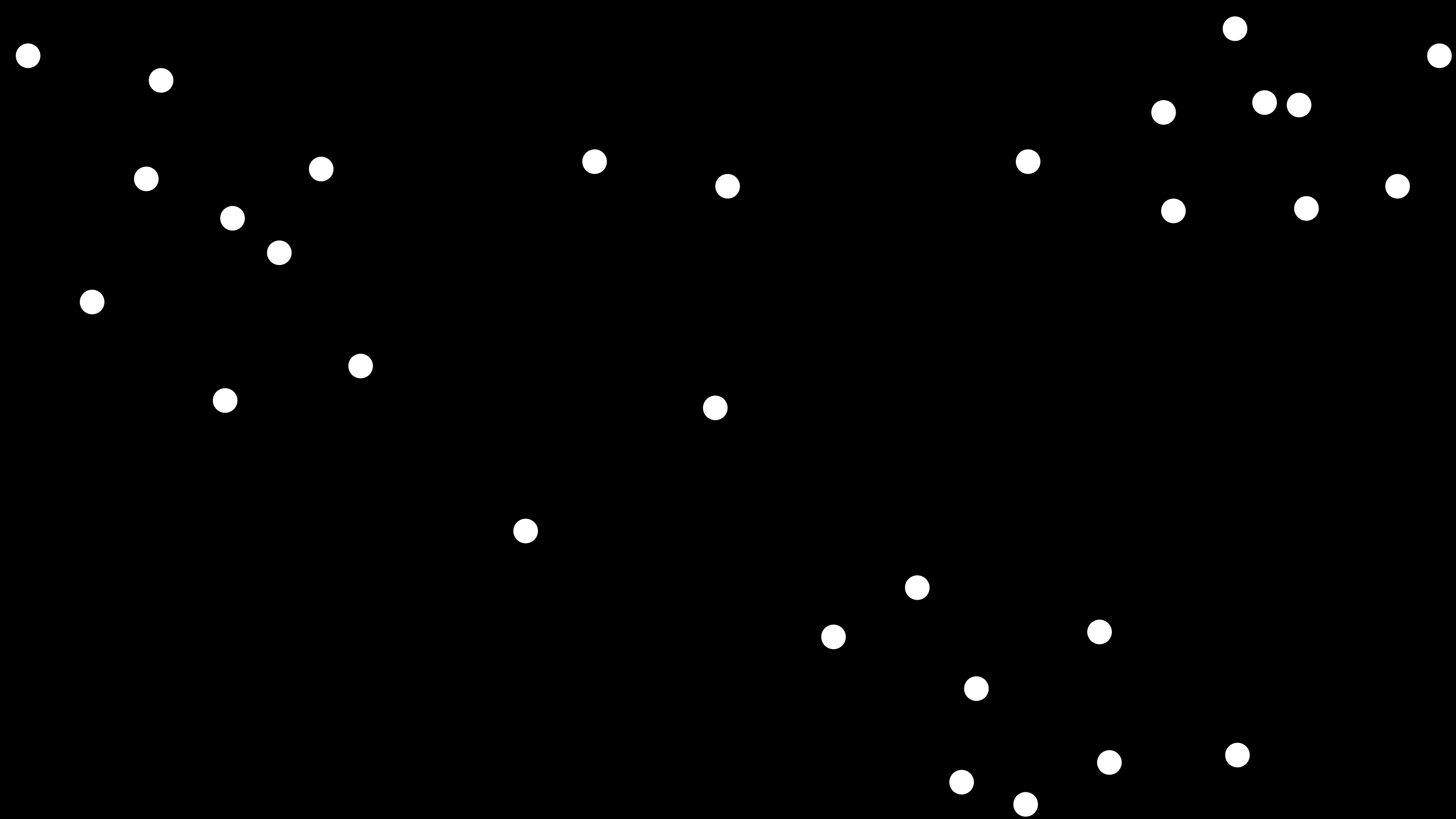
Fitting Gaussians



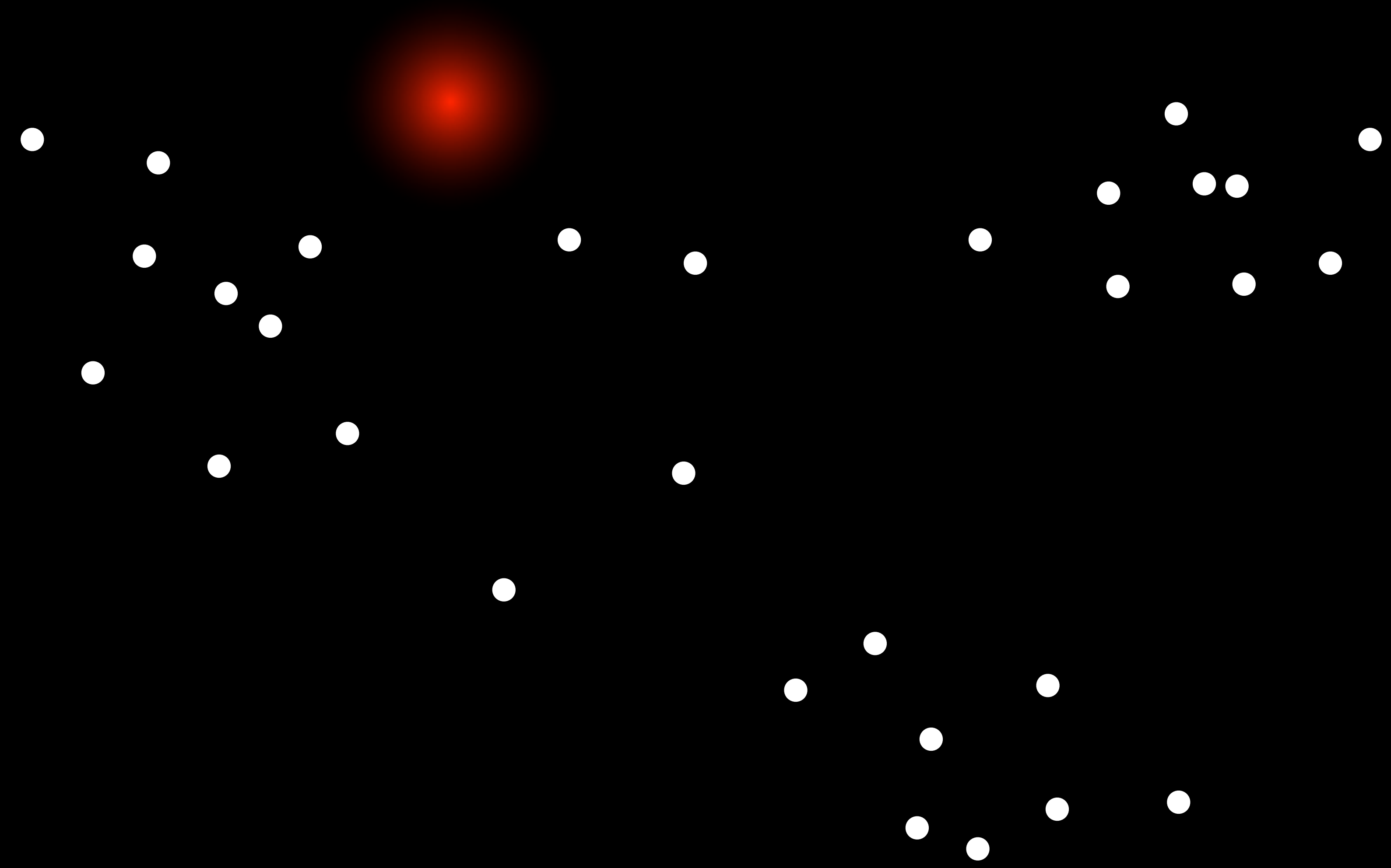
Fitting Gaussians



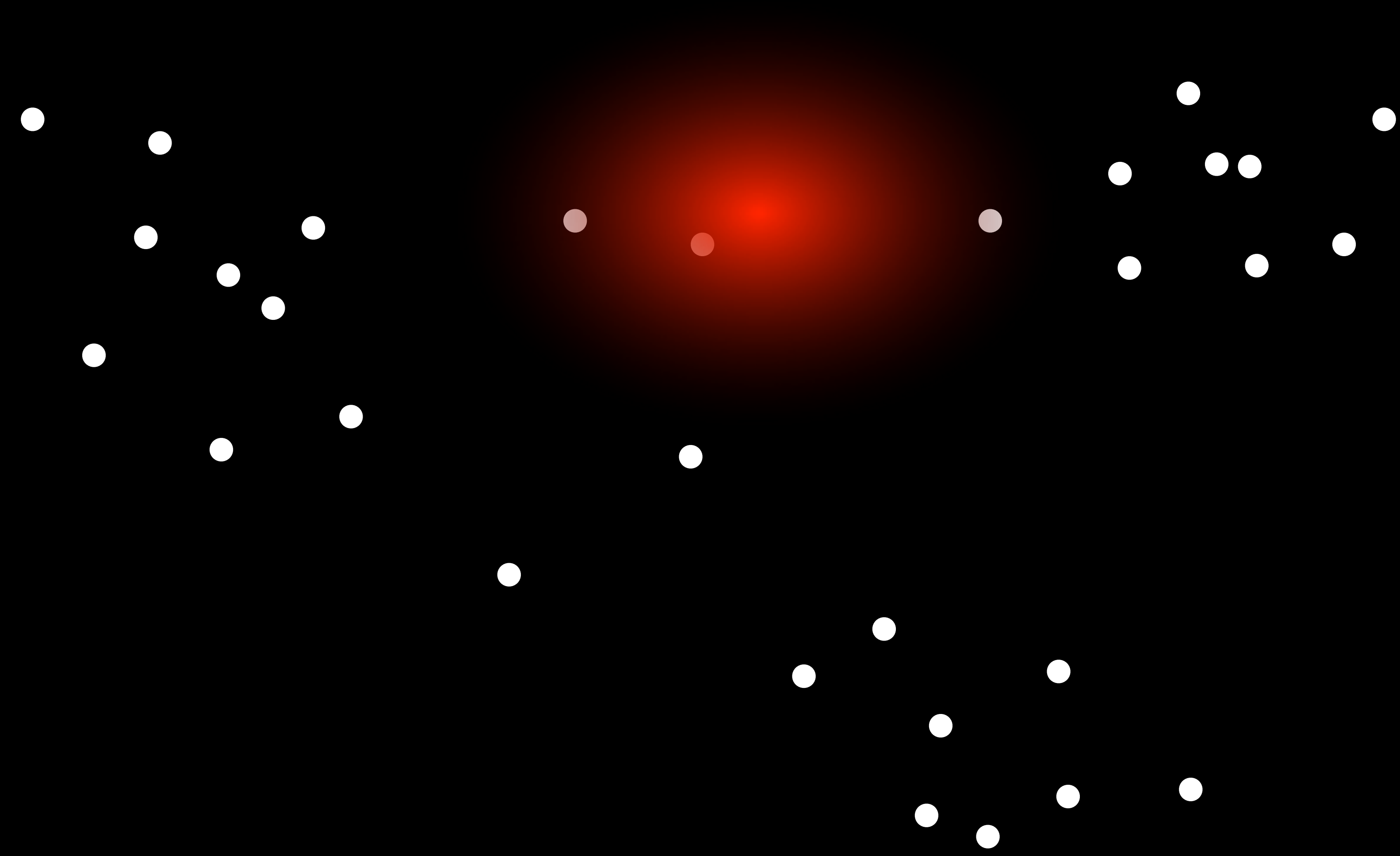
Fitting Gaussians



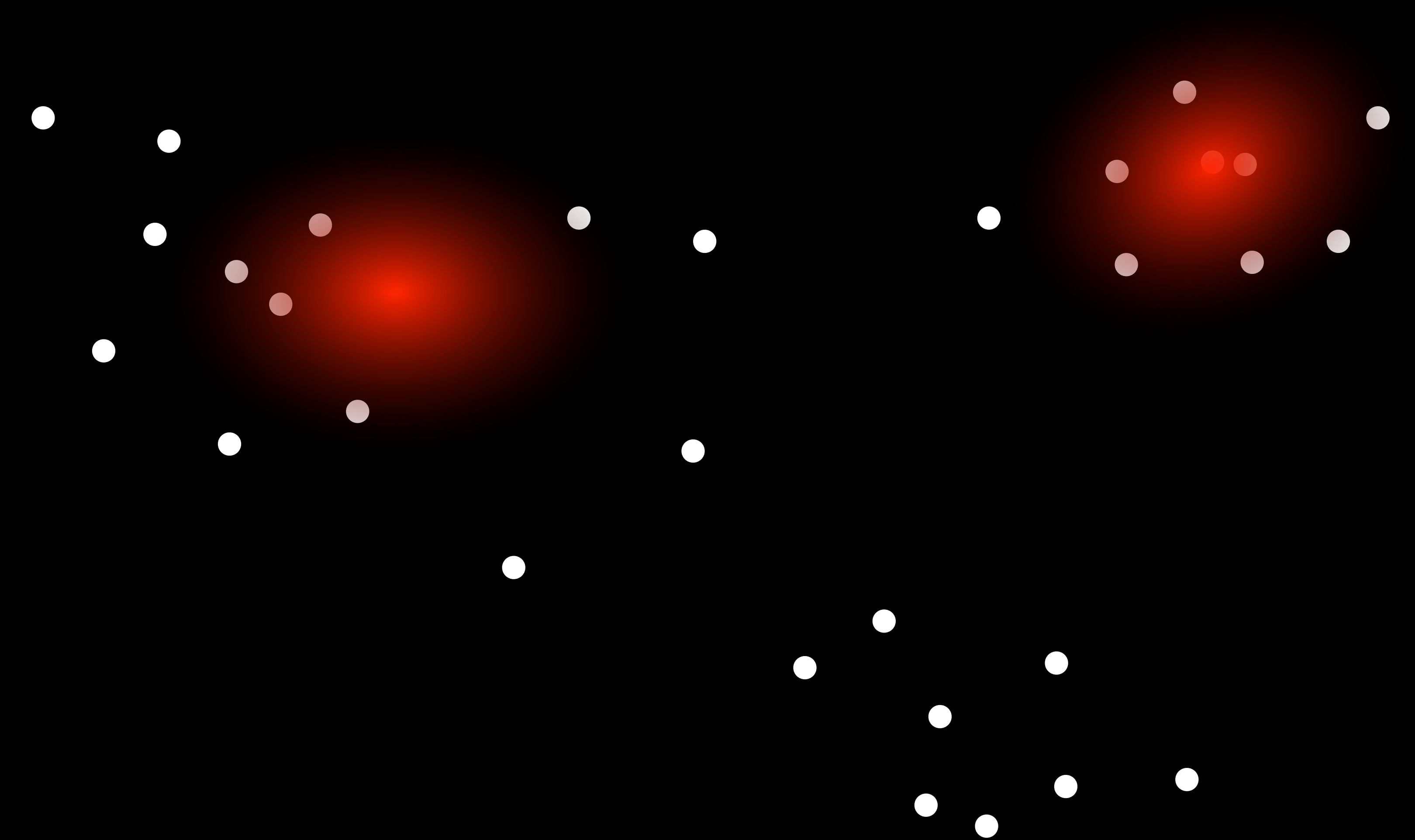
Fitting Gaussians



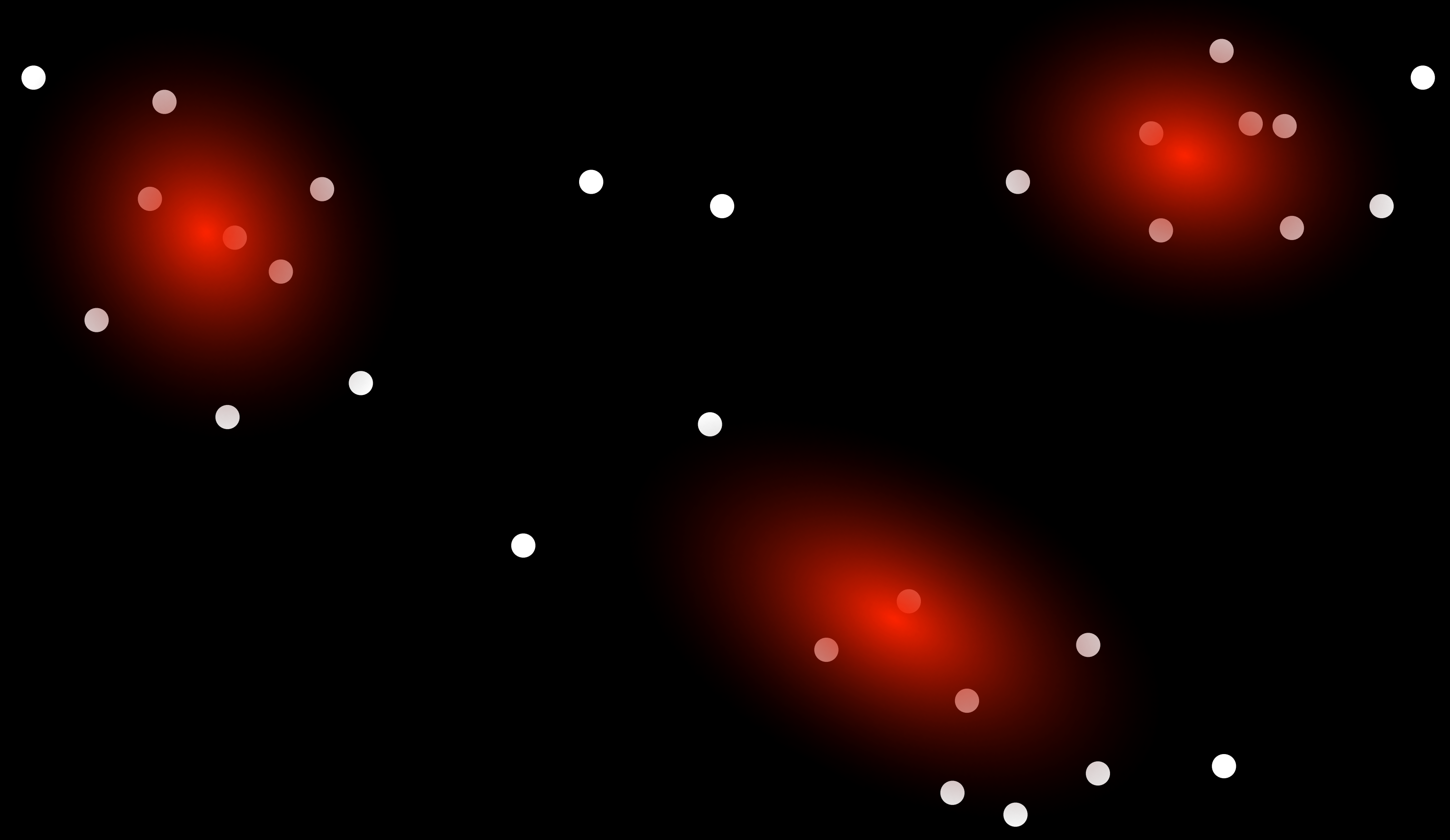
Fitting Gaussians



Fitting Gaussians



Fitting Gaussians



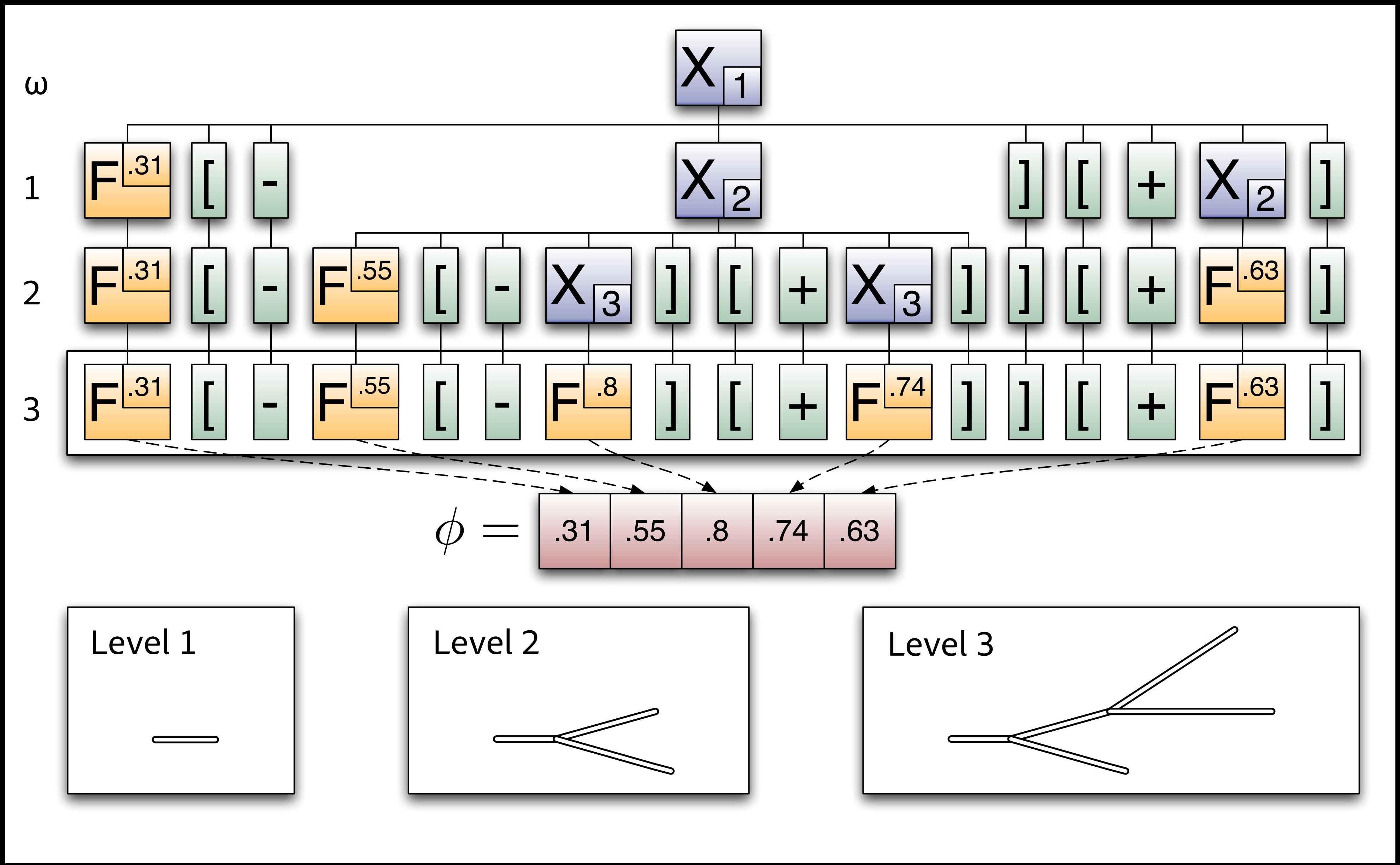
Bayesian Inference

- define model prior $\pi(\delta)$
- formulate likelihood function $L(I|\delta)$
- maximize $p(\delta|I) \propto L(I|\delta)\pi(\delta)$

RJMCMC

- start with random sample $\delta \sim \pi(\cdot)$
- dimension-preserving *diffusion* moves
- dimension-altering *jump* moves

Model Prior



$$\pi(\delta) \propto \prod_{s \in \delta} P\left(s|\text{parent}(s)\right) \prod_{t \in \delta_{\Rightarrow}} \prod_{i \in t} \Phi_{t(i)}(\phi_{t(i)}),$$

Likelihood Formulation

Take sketch/volume as input

- Render/voxelize current model
- Compute:

$$\log L(I|\delta) = -\frac{1}{2\sigma^2} \sum_{\vec{x} \in \mathcal{D}} d\big(I(\vec{x}), I_\delta(\vec{x})\big)^2$$

Likelihood Formulation

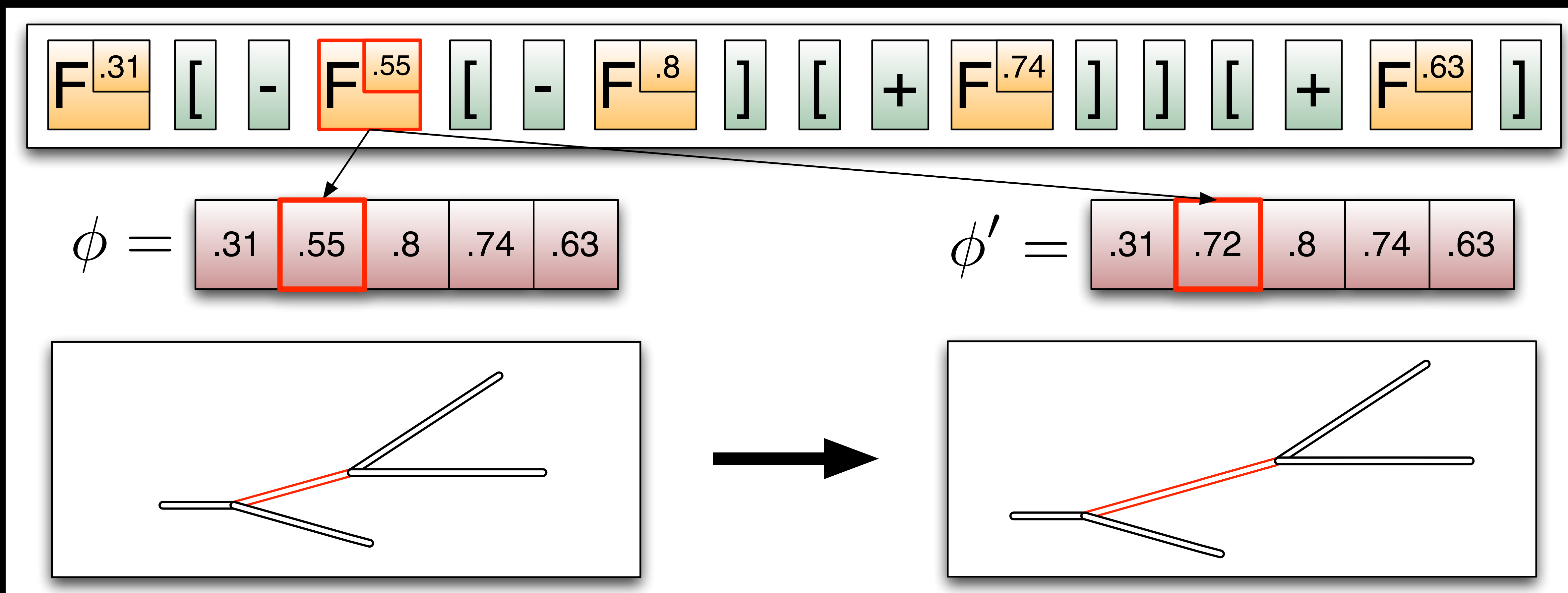
Take sketch/volume as input

- Render/voxelize current model
- Compute:

$$\log L(I|\delta) = -\frac{1}{2\sigma^2} \sum_{\vec{x} \in \mathcal{D}} d\big(I(\vec{x}), I_\delta(\vec{x})\big)^2$$

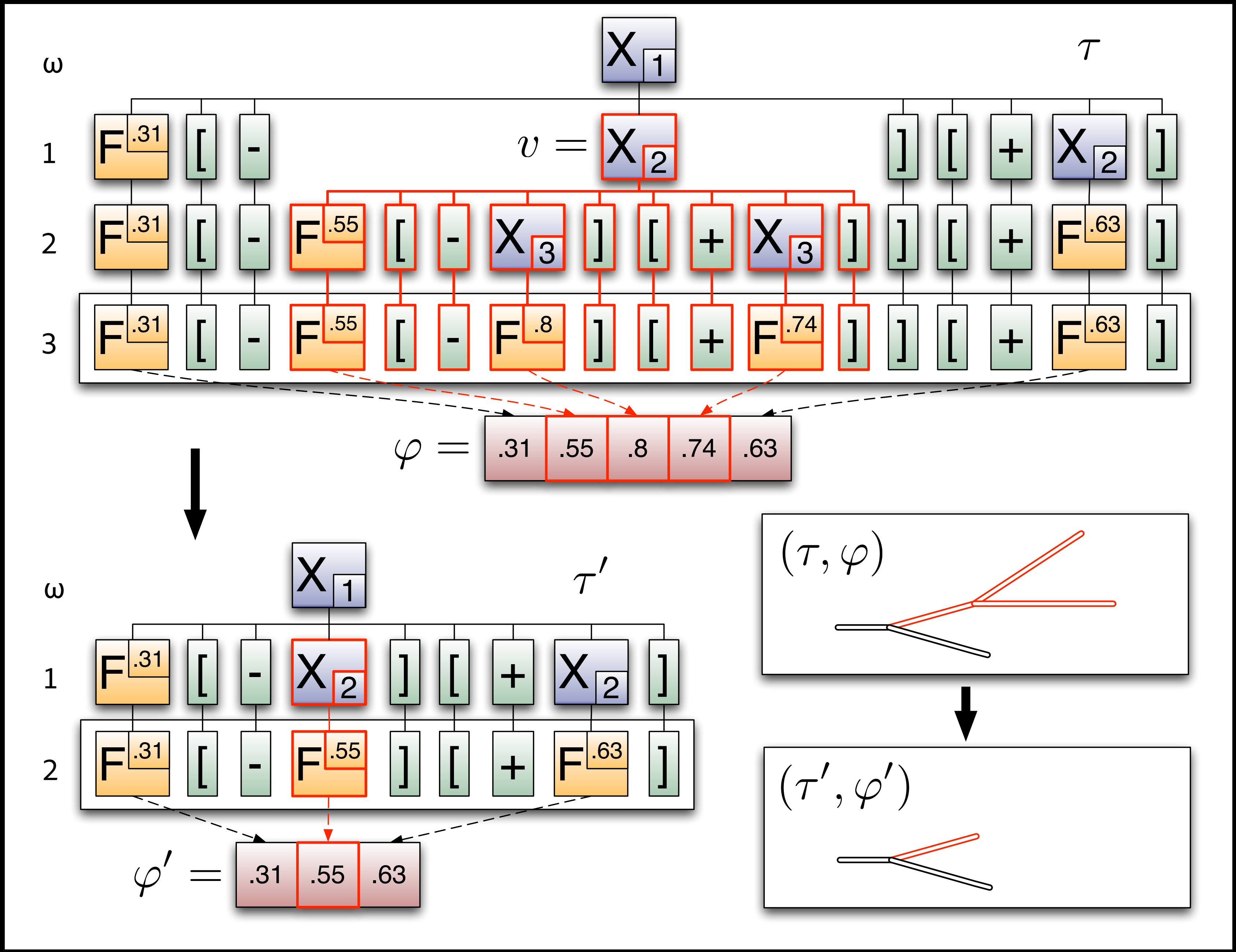
In principle, can use *any* cost function

Diffusion Moves



$$\alpha(\delta'|\delta) = \min\left\{1, \frac{p(\delta'|I)}{p(\delta|I)} \prod_i \frac{\Phi_{t(i)}(\phi_{t(i)})}{\Phi_{t(i)}(\phi'_{t(i)})}\right\} = \min\left\{1, \frac{L(I|\delta')}{L(I|\delta)}\right\}$$

Jump Moves


ω
τ
1
2
3
$v =$
$\varphi =$
.31
.55
.8
.74
.63
ω
τ'
1
2
$\varphi' =$
.31
.55
.63
(τ, φ)
(τ', φ')

Jump Moves

Theoretical requirements:

Jump Moves

Theoretical requirements:

- Reversibility

$$(n, \mathbf{x}_n) \to (m, \mathbf{x}_m) \iff (m, \mathbf{x}_m) \to (n, \mathbf{x}_n)$$

Jump Moves

Theoretical requirements:

- Reversibility

$$(n, \mathbf{x}_n) \to (m, \mathbf{x}_m) \iff (m, \mathbf{x}_m) \to (n, \mathbf{x}_n)$$

- Dimension-matching

$$f_{m \to n} : \mathcal{X}_m \times \mathcal{U}_{m,n} \to \mathcal{X}_n \times \mathcal{U}_{n,m}$$

with $f_{m \to n}$ deterministic, differentiable, invertible

$$f_{m \to n}\left([0.8, 0.4, 0.9]\right) = [0.8, 0.4, 0.9, 1, 0.5]$$

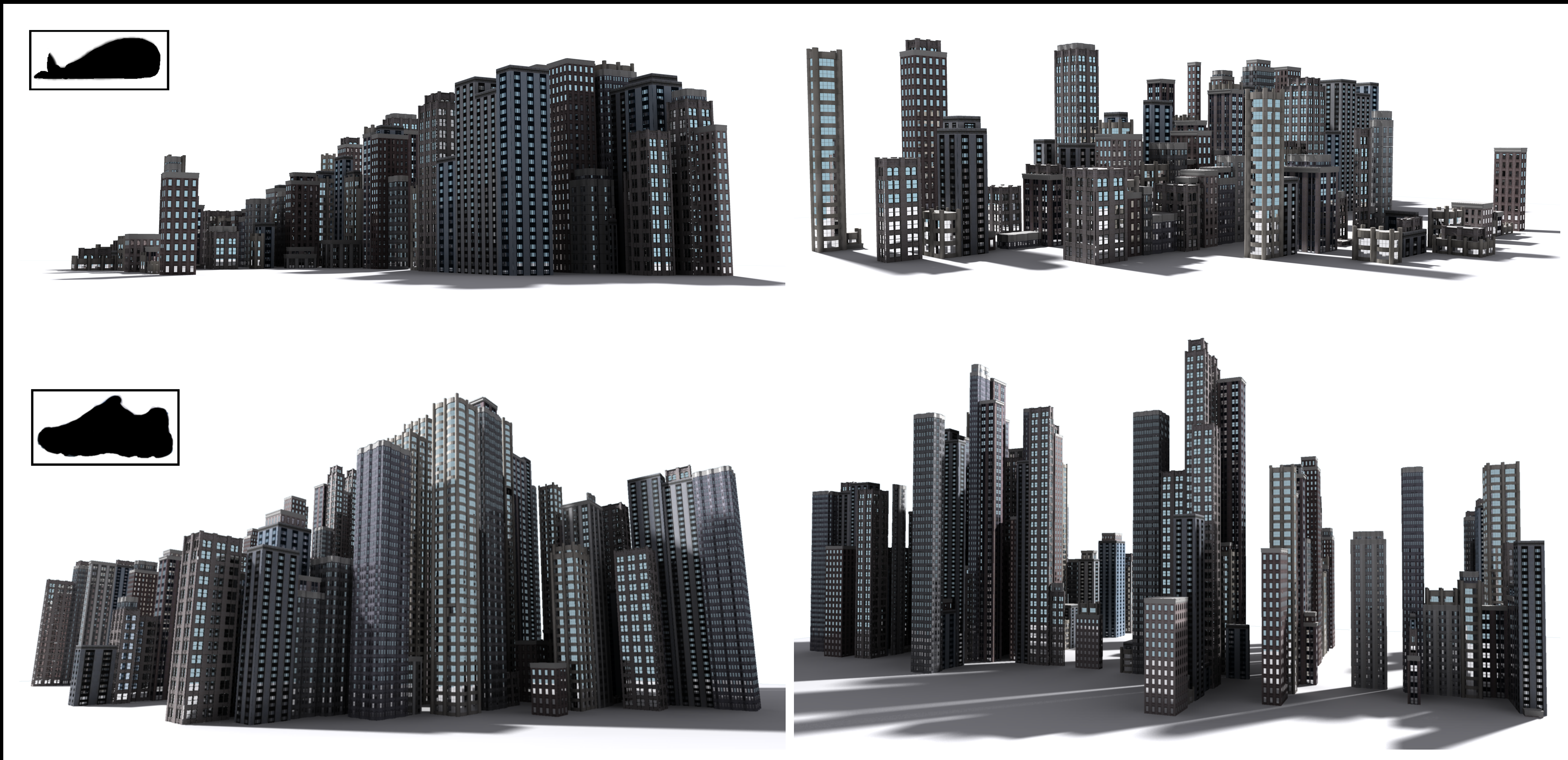
Jump Moves

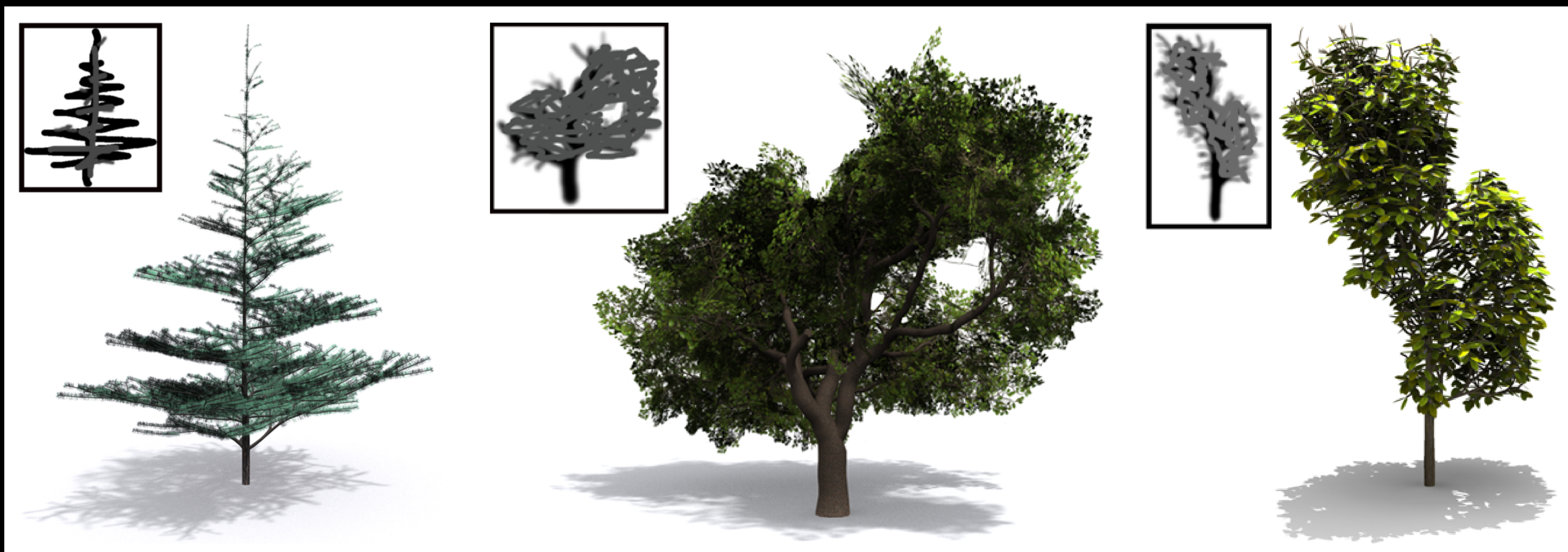
$$j(\tau'_v|\tau_v) = q_\tau(v) \prod_{s \in \tau'_v} P\left(s|\text{parent}(s)\right)$$

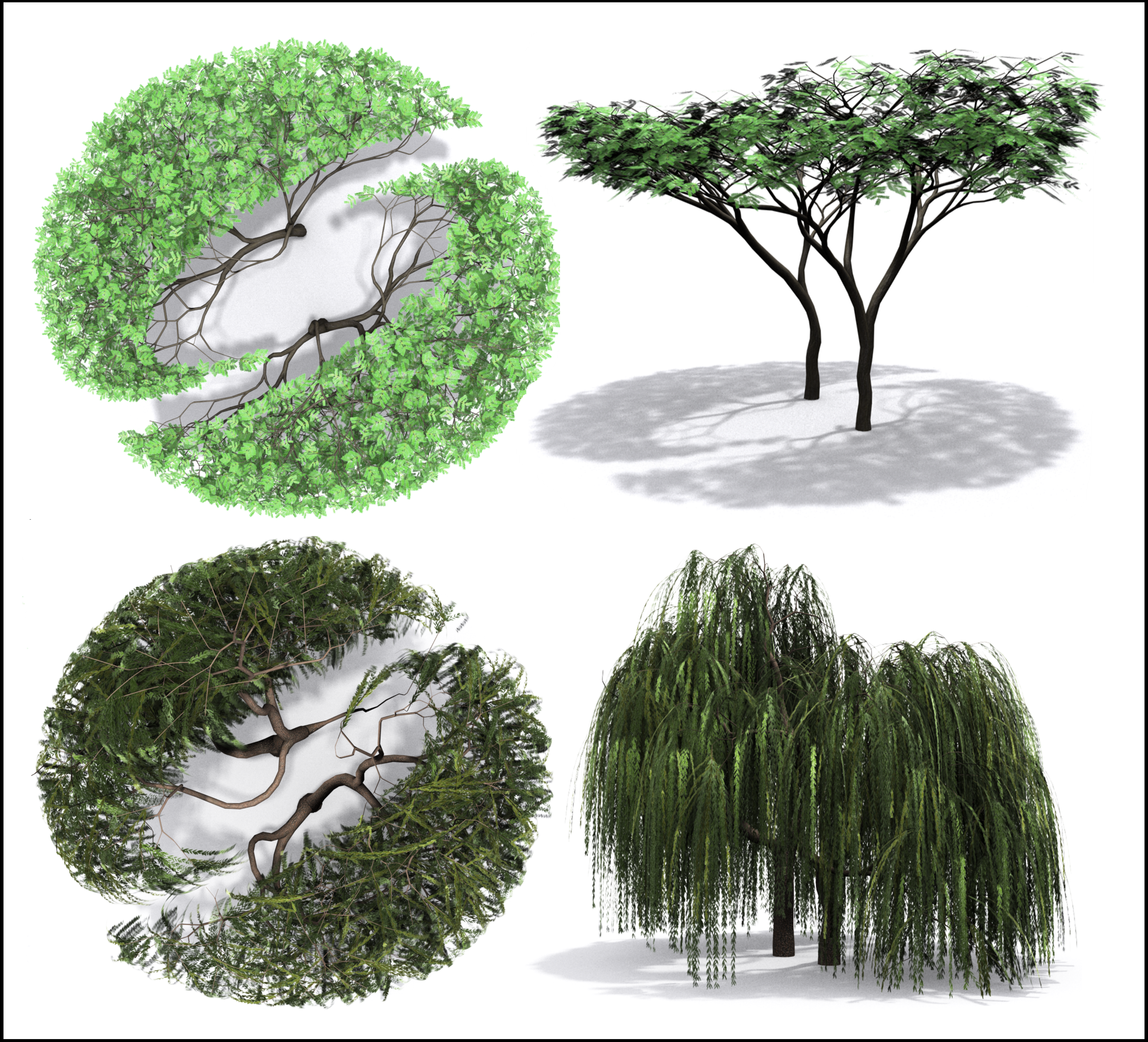
$$\alpha_{\delta \to \delta'} = \min\left\{1, \frac{p(\delta'|I)}{p(\delta|I)} \frac{j(\tau_v|\tau'_v)}{j(\tau'_v|\tau_v)} \frac{U_{[0,1]}(u')}{1} \mathcal{J}_{\delta \to \delta'}\right\}$$

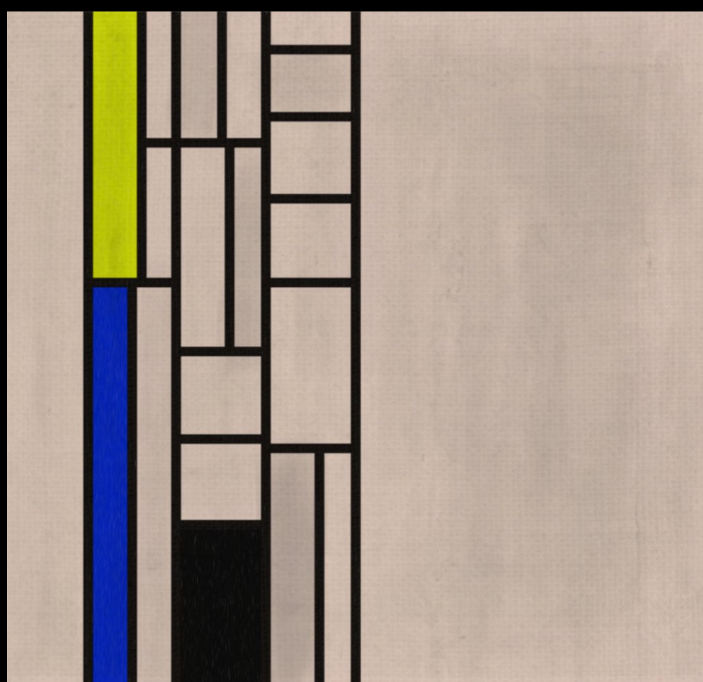
$$= \min\left\{1, \frac{L(I|\delta')\ \pi(\delta')}{L(I|\delta)\ \pi(\delta)} \frac{q_{\tau'}(v)\ \prod_{s \in \tau_v} P\left(s|\text{parent}(s)\right)}{q_\tau(v)\ \prod_{s \in \tau'_v} P\left(s|\text{parent}(s)\right)}\right\}$$

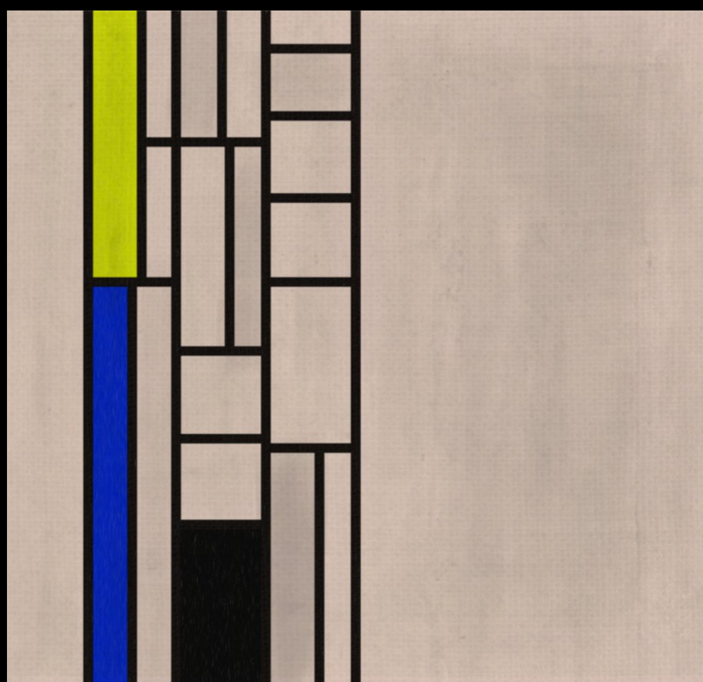
$$= \min\left\{1, \frac{q_{\tau'}(v)}{q_\tau(v)} \frac{L(I|\delta')}{L(I|\delta)} \frac{\prod_{t \in \tau'_v} \Phi_t\left(\varphi_t\right)}{\prod_{t \in \tau_v} \Phi_t\left(\varphi'_t\right)}\right\}$$

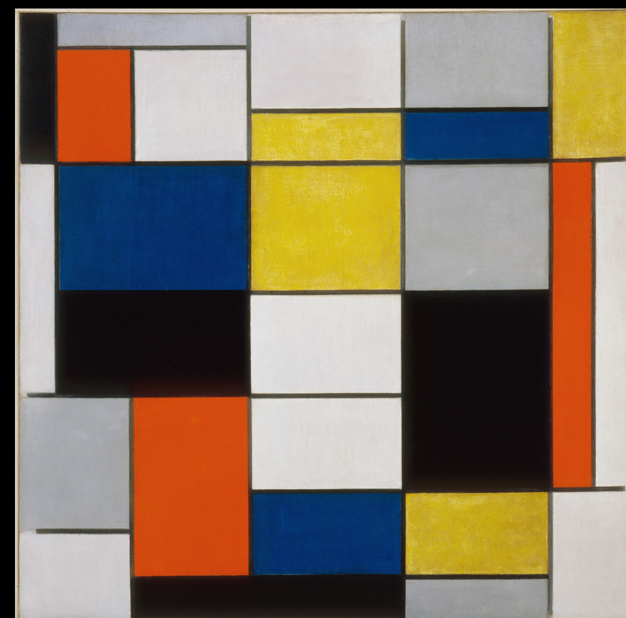


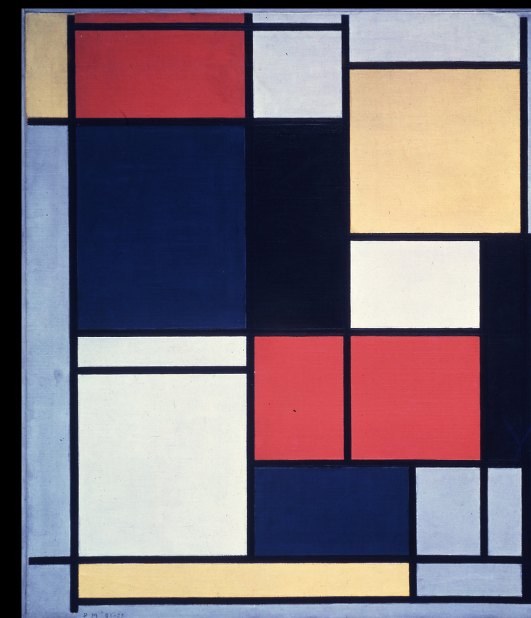


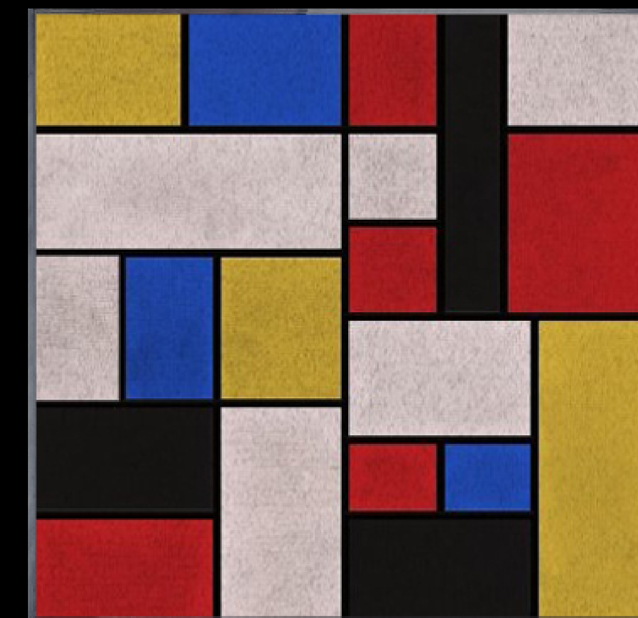


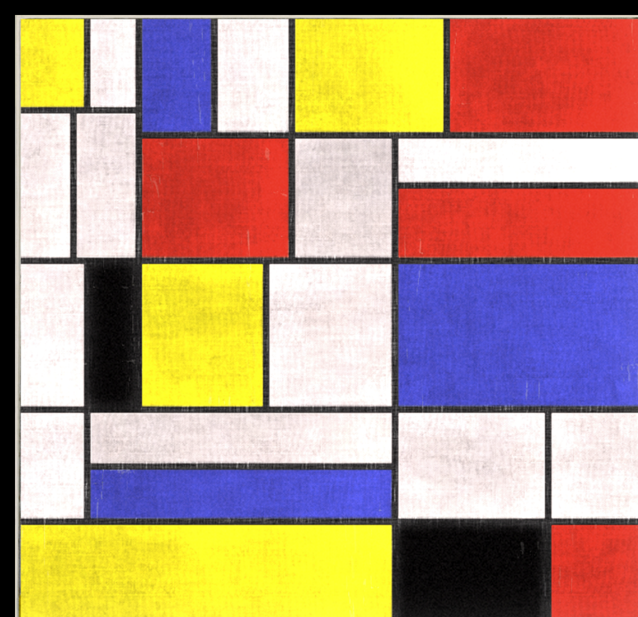


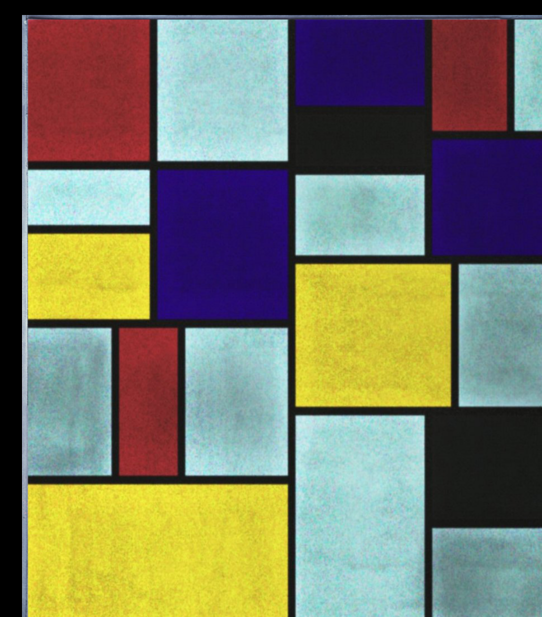


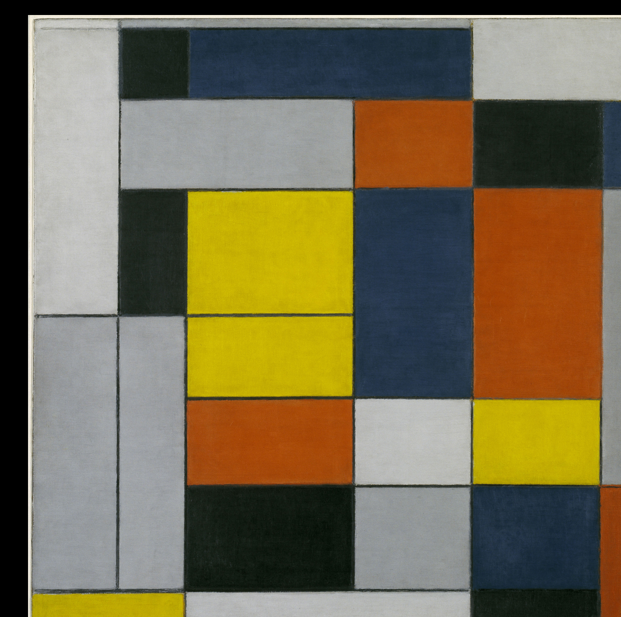












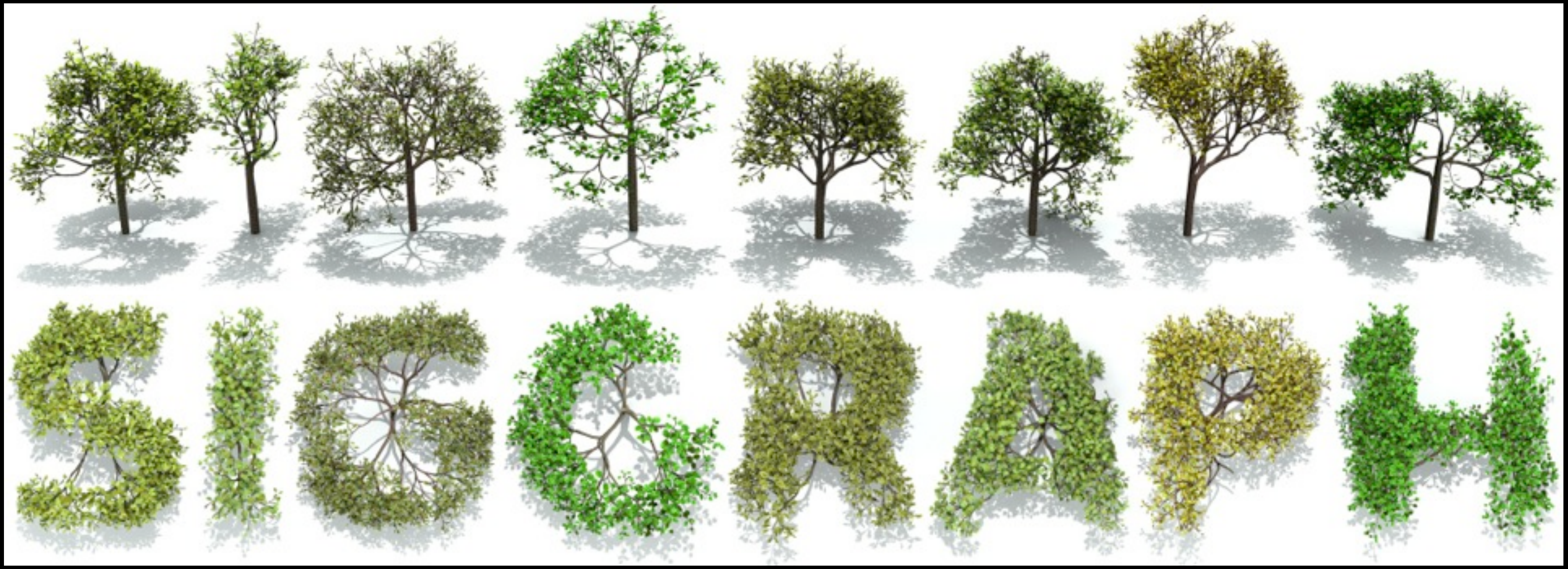
Limitations

Performance

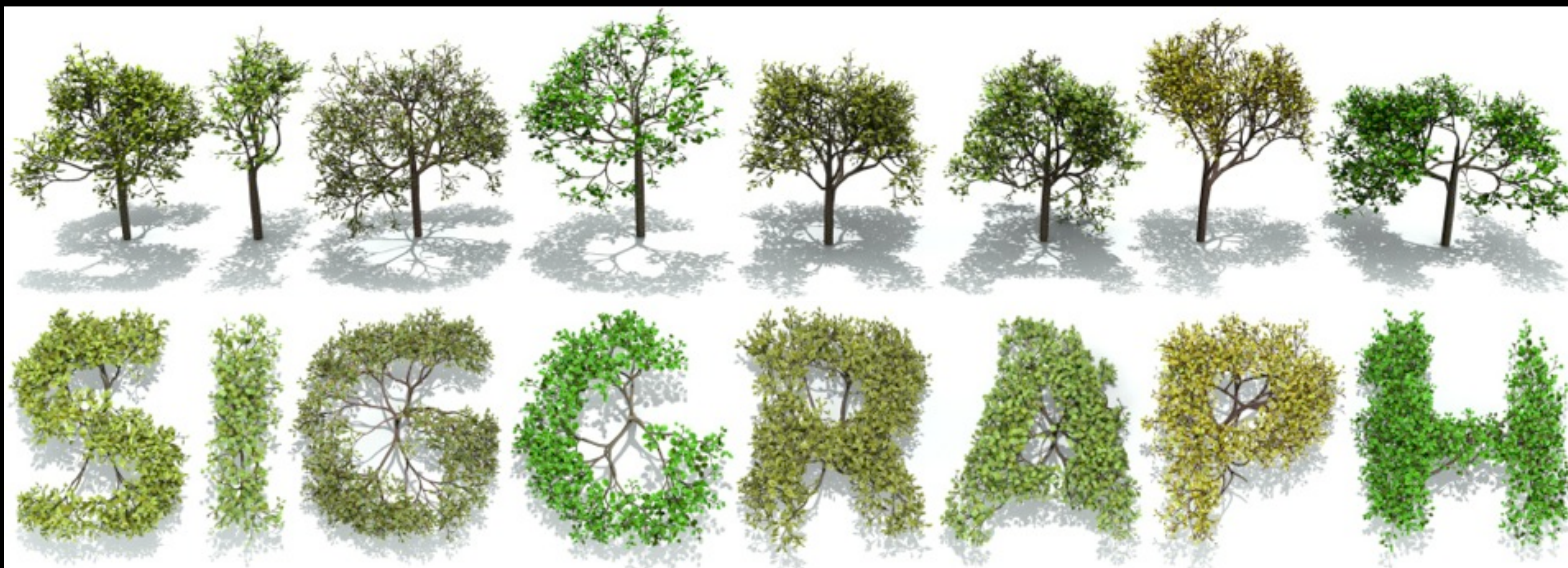
example	symbols	params	depth	vertices	N	time
City (6)	850	1,154	120	1,432	75k	14m
Young Oak (8)	12,152	18,937	15	176,520	110k	2h
Acacia (9)	21,176	29,909	29	50,862	265k	4h
Willow (9)	70,754	164,823	34	166,328	385k	10h
Conifer (10)	31,319	35,639	27	221,941	350k	9h
Old Oak (10)	53,624	71,738	30	66,191	560k	6h
Poplar (10)	11,019	12,836	32	193,016	90k	2h
Building (11)	9,673	16,805	28	1,660	850k	30m
Mondrian (15)	74	134	21	38	6k	5s

- Tempered transitions [Neal ’94], parallel tempering [Geyer ’91], delayed rejection [Tierney & Mira ’99]
- Sequential & data-driven MCMC [Tu & Zhu ’02]
- Coupling from the past [Propp & Wilson ’96]

graphics.stanford.edu/projects/mpm



graphics.stanford.edu/projects/mpm



Questions?