Compositional Sculpting of Iterative Generative Processes

Timur Garipov   Sebastiaan De Peuter   Ge Yang

Vikas Garg   Samuel Kaski   Tommi Jaakkola

Aalto University School of Science

The University of Manchester

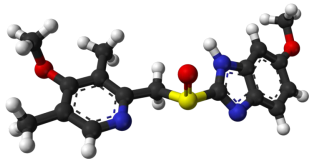
Composition of generative models

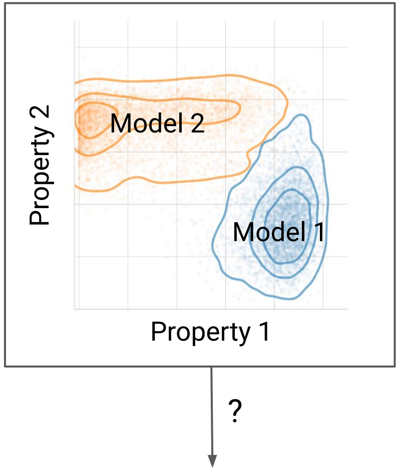
Composition: construction of complex models from simpler building blocks

- Large-scale general-purpose pre-training is becoming ubiquitous
- Need to re-use and adapt **pre-trained models** for new tasks
- Combining knowledge from multiple sources (models, datasets)

Composition → adjustment of sampling distribution

- Applications such as multi-objective molecule generation
- Need to explore trade-offs between multiple criteria





(Property 1) AND (Property 2)

Prior work: model composition

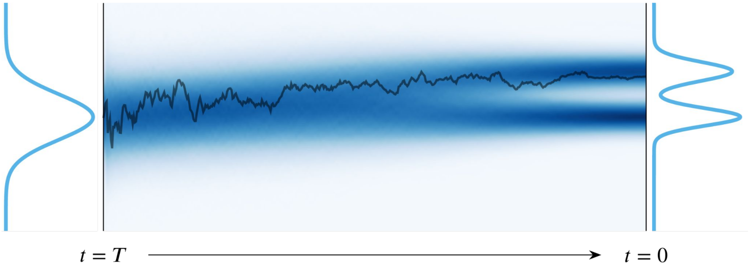
	Models	Composition Operations	Sampling Algorithm
[Hinton, Neural Computation 2002] [Du et al, NeurIPS 2020]	Energy-based models (EBMs) $p_i(x) \propto \exp(-E_i(x;\theta))$	Principle: energy-function arithmetic Product: $\frac{1}{Z} p_1(x)\, p_2(x)$ Negation: $\frac{1}{Z} \frac{p_1(x)}{\left(p_2(x)\right)^\gamma}$	MCMC Langevin dynamics

Iterative Generative Processes

Diffusion model generates trajectories $\tau = \left(x_T \to \{x_t\}_{t=0}^{T} \to x_0\right)$ with terminal state distribution $p(x_0)$ by following a backward SDE

$$dx_t = \left[f_t(x_t) - g_t^2 \nabla_{x_t} \log p_t(x_t)\right] dt + g_t \, d\overline{w}_t,$$

corresponding to a forward noising process $dx_t = f_t(x_t)\, dt + g_t \, dw_t$



GFlowNet generates trajectories $\tau = \left(s_0 \to \ldots \to s_{n-1} \to x\right)$ with terminal distribution $p(x) = \frac{R(x)}{Z}$ by following a forward policy

$$p_F(s_t \mid s_{t-1}; \theta)$$

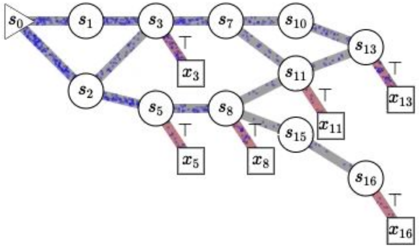


Image source: yoshuabengio.org/2022/03/05/generative-flow-networks

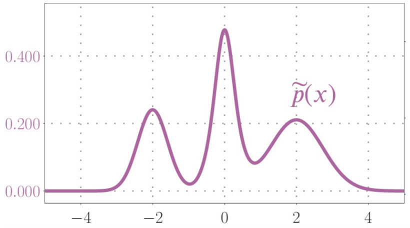
Prior work: model composition

	Models	Composition Operations	Sampling Algorithm
[Hinton, Neural Computation 2002] [Du et al, NeurIPS 2020]	Energy-based models (EBMs) $p_i(x) \propto \exp(-E_i(x;\theta))$	Principle: energy-function arithmetic Product: $\frac{1}{Z} p_1(x)\, p_2(x)$ Negation: $\frac{1}{Z} \frac{p_1(x)}{\left(p_2(x)\right)^\gamma}$	MCMC Langevin dynamics
Challenge: iterative generative processes (Diffusion models & GFlowNets) impose delicate balance conditions			
[Liu et al, ECCV 2022] [Du et al, ICML 2023]	Diffusion models $p_i(x): s_{i,t}(x_t;\theta) \approx \nabla_{x_t} \log p_{i,t}(x_t)$	Principle: score-function arithmetic Product: $\frac{1}{Z} p_1(x)\, p_2(x)$ Negation: $\frac{1}{Z} \frac{p_1(x)}{\left(p_2(x)\right)^\gamma}$	Diffusion sampling + annealed MCMC

Observational guidance

Prior

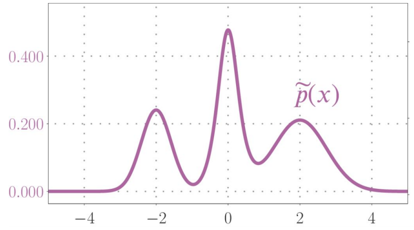
$$\widetilde{p}(x)$$



Observational guidance

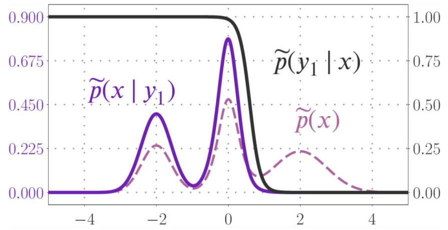
Prior

$$\widetilde{p}(x)$$



Condition on observation y_1

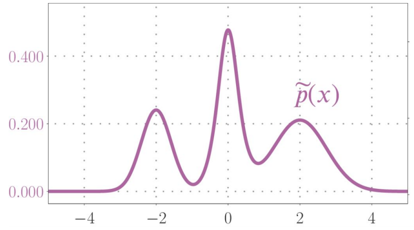
$$\underbrace{\widetilde{p}(x \mid y_1)}_{\text{Posterior}} \propto \underbrace{\widetilde{p}(x)}_{\text{Prior}} \underbrace{\widetilde{p}(y_1 \mid x)}_{\text{Likelihood}}$$



Observational guidance

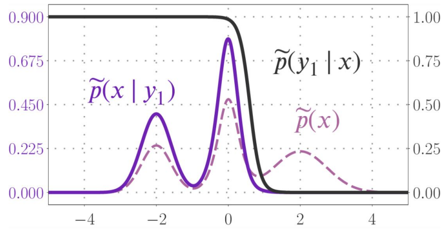
Prior

$$\widetilde{p}(x)$$



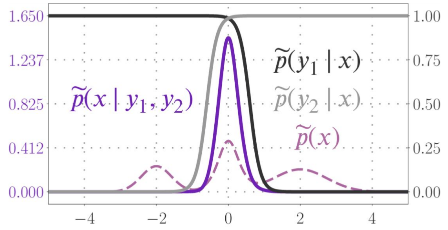
Condition on observation y_1

$$\overbrace{\widetilde{p}(x \mid y_1)}^{\text{Posterior}} \propto \overbrace{\widetilde{p}(x)}^{\text{Prior}}\,\overbrace{\widetilde{p}(y_1 \mid x)}^{\text{Likelihood}}$$

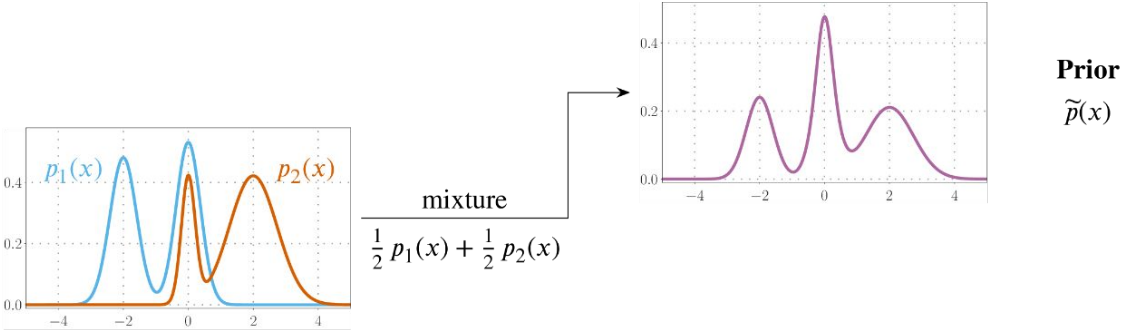


Condition on observation y_1 and y_2

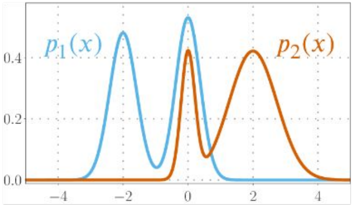
$$\widetilde{p}(x \mid y_1, y_2) \propto \widetilde{p}(x)\,\underbrace{\widetilde{p}(y_1 \mid x)\widetilde{p}(y_2 \mid x)}_{\text{Higher density on } x \text{ matching } y_1 \text{ and } y_2.}$$



Compositional Sculpting

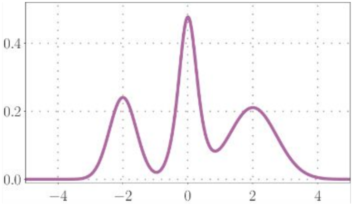


Compositional Sculpting



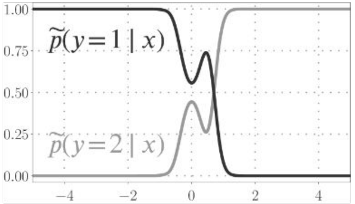
mixture

$\frac{1}{2}\, p_1(x) + \frac{1}{2}\, p_2(x)$



Prior

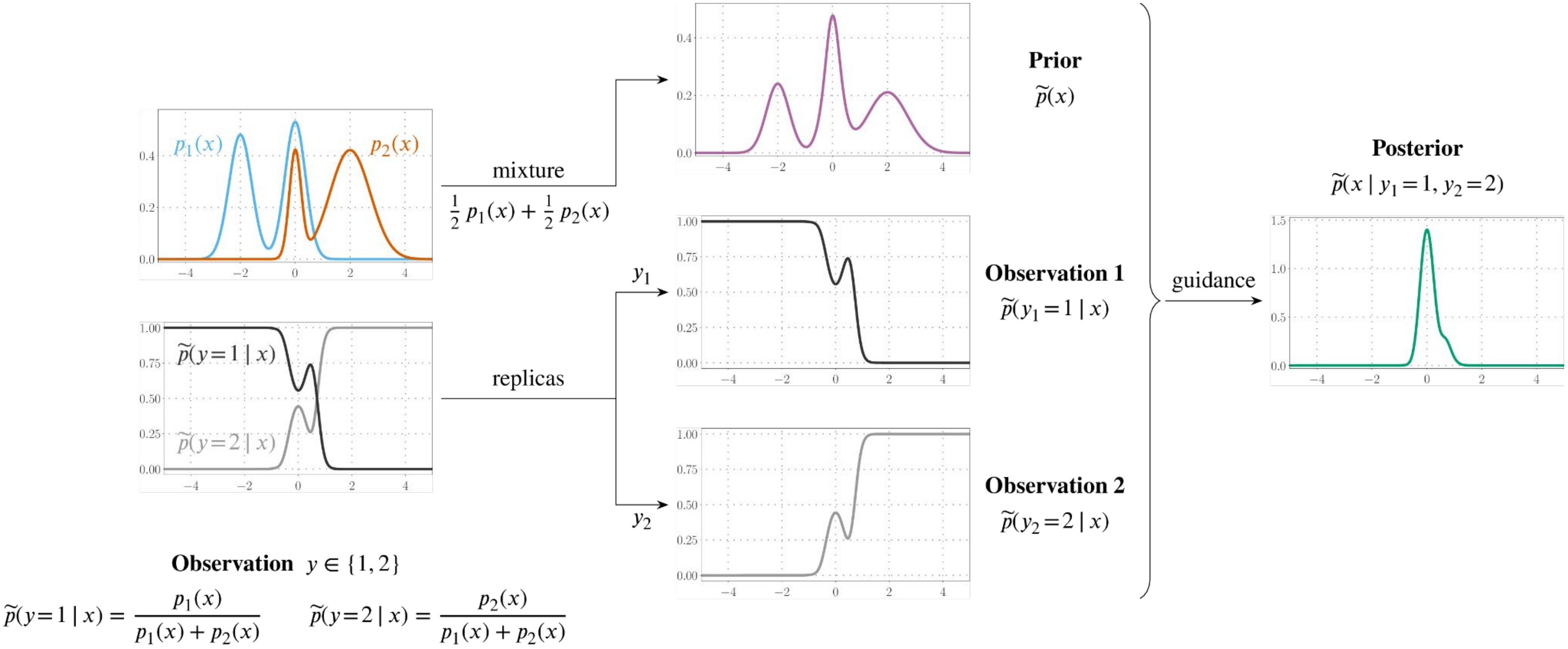
$\widetilde{p}(x)$



Observation $y \in \{1, 2\}$

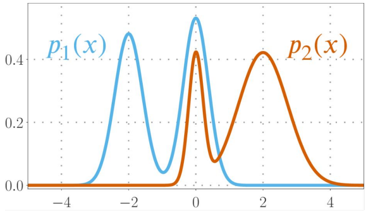
$$\widetilde{p}(y=1 \mid x) = \frac{p_1(x)}{p_1(x) + p_2(x)} \qquad \widetilde{p}(y=2 \mid x) = \frac{p_2(x)}{p_1(x) + p_2(x)}$$

Compositional Sculpting



Compositional Sculpting: operations

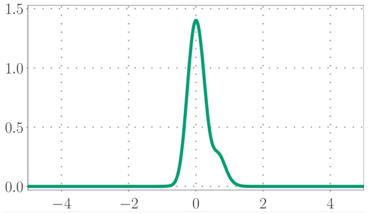
Given: 2 distributions $p_1(x)$, $p_2(x)$



Prior

$$\widetilde{p}(x) = \frac{1}{2}p_1(x) + \frac{1}{2}p_2(x)$$

$(p_1 \otimes p_2)(x)$



Observations

$$\widetilde{p}(y_k = i \mid x) = \frac{p_i(x)}{p_1(x) + p_2(x)}, \qquad i \in \{1, 2\}$$

Posterior (composition)

$$\widetilde{p}(x \mid y_1 = i, y_2 = j) = \frac{p_i(x)p_j(x)}{p_1(x) + p_2(x)}$$

"Harmonic Mean"

$$(p_1 \otimes p_2)(x) \;=\; \widetilde{p}(x \mid y_1 = 1, y_2 = 2) \;\propto\; \frac{p_1(x)p_2(x)}{p_1(x) + p_2(x)}$$

Compositional Sculpting: operations

Given: 2 distributions $p_1(x)$, $p_2(x)$

Prior

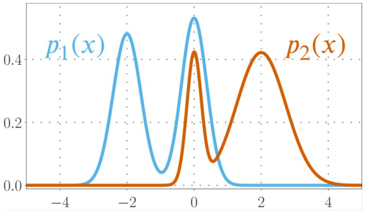
$$\widetilde{p}(x) = \frac{1}{2}p_1(x) + \frac{1}{2}p_2(x)$$

Observations

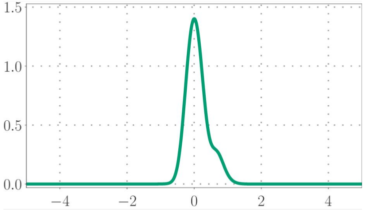
$$\widetilde{p}(y_k = i \mid x) = \frac{p_i(x)}{p_1(x) + p_2(x)}, \quad i \in \{1, 2\}$$

Posterior (composition)

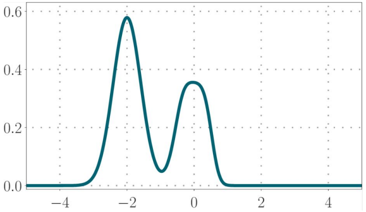
$$\widetilde{p}(x \mid y_1 = i, y_2 = j) = \frac{p_i(x)p_j(x)}{p_1(x) + p_2(x)}$$



$(p_1 \otimes p_2)(x)$



$(p_1 \text{◑} p_2)(x)$



"Harmonic Mean"

$$(p_1 \otimes p_2)(x) = \widetilde{p}(x \mid y_1 = 1, y_2 = 2) \propto \frac{p_1(x)p_2(x)}{p_1(x) + p_2(x)}$$

"Contrast"

$$(p_1 \text{◑} p_2)(x) = \widetilde{p}(x \mid y_1 = 1, y_2 = 1) \propto \frac{\left(p_1(x)\right)^2}{p_1(x) + p_2(x)}$$

Compositional Sculpting: operations

Given: 2 distributions $p_1(x)$, $p_2(x)$

Prior

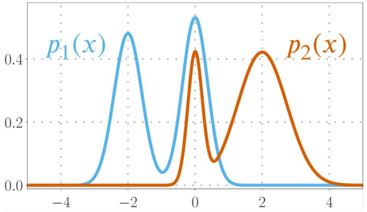
$$\widetilde{p}(x) = \frac{1}{2}p_1(x) + \frac{1}{2}p_2(x)$$

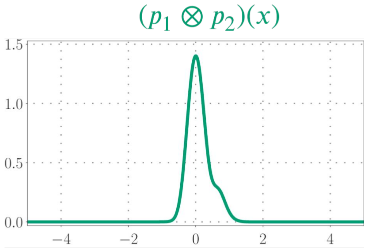
Observations

$$\widetilde{p}(y_k = i \mid x) = \frac{p_i(x)}{p_1(x) + p_2(x)}, \quad i \in \{1, 2\}$$

Posterior (composition)

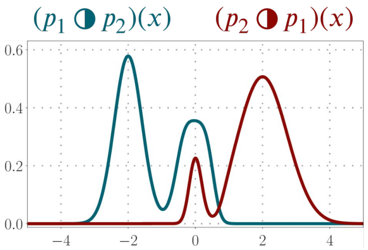
$$\widetilde{p}(x \mid y_1 = i, y_2 = j) = \frac{p_i(x)p_j(x)}{p_1(x) + p_2(x)}$$





"Harmonic Mean"

$$(p_1 \otimes p_2)(x) = \widetilde{p}(x \mid y_1 = 1, y_2 = 2) \propto \frac{p_1(x)p_2(x)}{p_1(x) + p_2(x)}$$



"Contrast"

$$(p_1 ◑ p_2)(x) = \widetilde{p}(x \mid y_1 = 1, y_2 = 1) \propto \frac{\left(p_1(x)\right)^2}{p_1(x) + p_2(x)}$$

$$(p_2 ◑ p_1)(x) = \widetilde{p}(x \mid y_1 = 2, y_2 = 2) \propto \frac{\left(p_2(x)\right)^2}{p_1(x) + p_2(x)}$$

Distribution composition

	Models	Composition Operations	Sampling Algorithm
[Hinton, Neural Computation 2002] [Du et al, NeurIPS 2020]	Energy-based models (EBMs) $p_i(x) \propto \exp(-E_i(x;\theta))$	Principle: energy-function arithmetic Product: $\frac{1}{Z}\, p_1(x)\, p_2(x)$ Negation: $\frac{1}{Z}\, \frac{p_1(x)}{\left(p_2(x)\right)^{\gamma}}$	MCMC Langevin dynamics
Challenge: iterative generative processes (Diffusion models & GFlowNets) impose delicate balance conditions			
[Liu et al, ECCV 2022] [Du et al, ICML 2023]	Diffusion models $p_i(x):\ s_{i,t}(x_t;\theta) \approx \nabla_{x_t} \log p_{i,t}(x_t)$	Principle: score-function arithmetic Product: $\frac{1}{Z}\, p_1(x)\, p_2(x)$ Negation: $\frac{1}{Z}\, \frac{p_1(x)}{\left(p_2(x)\right)^{\gamma}}$	Diffusion sampling + annealed MCMC
Compositional sculpting (ours)	Diffusion models $p_i(x):\ s_{i,t}(x_t;\theta) \approx \nabla_{x_t} \log p_{i,t}(x_t)$ GFlowNets $p_i(x):\ p_{i,F}(s_{t+1} \mid s_t;\theta)$	Principle: mixture & conditional generative processes **Harmonic Mean**: $\frac{1}{Z}\, \frac{p_1(x)\, p_2(x)}{p_1(x) + p_2(x)}$ **Contrast**: $\frac{1}{Z}\, \frac{\left(p_1(x)\right)^2}{p_1(x) + p_2(x)}$ (+) **other operations** in the paper	Diffusion mixture + classifier guidance GFlowNet mixture + classifier guidance

Composition: mixture & guidance

Diffusion

$$\underbrace{\begin{array}{rccc} \text{base models} & p_1(x_0) & \dots & p_m(x_0) \\ \text{score functions} & s_{1,t}(x_t) & \dots & s_{m,t}(x_t) \\ \text{observations} & \widetilde{p}(y_k = i \mid x_0) & = & \frac{p_i(x_0)}{\sum_{j=1}^m p_j(x_0)} \end{array}}$$

$$\Downarrow$$

$$\text{composite model} \quad \widetilde{p}(x_0 \mid y_1, \dots, y_n)$$

$$s_t(x_t, y_1, \dots, y_n)$$

$$\overbrace{\underbrace{\left(\sum_{i=1}^m p(y=i \mid x_t) s_{i,t}(x_t)\right)}_{\text{mixture}} + \underbrace{\nabla_{x_t} \log p(y_1, \dots, y_n \mid x_t)}_{\text{guidance}}}$$

GFlowNets

$$\underbrace{\begin{array}{rccc} \text{base models} & p_1(x) & \dots & p_m(x) \\ \text{forward policies} & p_{1,F}(s' \mid s) & \dots & p_{m,F}(s' \mid s) \\ \text{observations} & \widetilde{p}(y_k = i \mid x) & = & \frac{p_i(x)}{\sum_{j=1}^m p_j(x)} \end{array}}$$

$$\Downarrow$$

$$\text{composite model} \quad \widetilde{p}(x \mid y_1, \dots, y_n)$$

$$p_F(s' \mid s, y_1, \dots y_n)$$

$$\overbrace{\underbrace{\left(\sum_{i=1}^m p(y=i \mid s) p_{i,F}(s' \mid s)\right)}_{\text{mixture}} \cdot \underbrace{\frac{p(y_1, \dots, y_n \mid s')}{p(y_1, \dots, y_n \mid s)}}_{\text{guidance}}}$$

Classifier training (diffusion)

Full probabilistic model $\widetilde{p}(\{x_t\}_{t=0}^{T})\widetilde{p}(y_1 \mid x_0)\widetilde{p}(y_2 \mid x_0)$

Compositions $\widetilde{p}(x \mid y_1 = i, y_2 = j)$

Mixtures & guidance require $\widetilde{p}(y \mid x_t)$, $\nabla_{x_t} \log \widetilde{p}(y_1, y_2 \mid x_t)$

Method: train classifier $\widetilde{Q}_\phi(y_1, y_2 \mid x_t) \approx \widetilde{p}(y_1, y_2 \mid x_t)$

Classifier training (diffusion)

Full probabilistic model $\widetilde{p}(\{x_t\}_{t=0}^{T})\widetilde{p}(y_1 \mid x_0)\widetilde{p}(y_2 \mid x_0)$

Compositions $\widetilde{p}(x \mid y_1 = i, y_2 = j)$

Mixtures & guidance require $\widetilde{p}(y \mid x_t)$, $\nabla_{x_t} \log \widetilde{p}(y_1, y_2 \mid x_t)$

Method: train classifier $\widetilde{Q}_\phi(y_1, y_2 \mid x_t) \approx \widetilde{p}(y_1, y_2 \mid x_t)$

Terminal classifier $(t = 0)$ $\widetilde{Q}_\phi(y \mid x_0) \approx \widetilde{p}(y \mid x_0)$

Training data $(\hat{x}_0, \hat{y}_1) \sim \widetilde{p}(x_0, y_1)$

Classifier training (diffusion)

Full probabilistic model $\widetilde{p}(\{x_t\}_{t=0}^{T})\widetilde{p}(y_1 \mid x_0)\widetilde{p}(y_2 \mid x_0)$

Compositions $\widetilde{p}(x \mid y_1 = i, y_2 = j)$

Mixtures & guidance require $\widetilde{p}(y \mid x_t)$, $\nabla_{x_t} \log \widetilde{p}(y_1, y_2 \mid x_t)$

Method: train classifier $\widetilde{Q}_\phi(y_1, y_2 \mid x_t) \approx \widetilde{p}(y_1, y_2 \mid x_t)$

Terminal classifier $(t = 0)$ $\widetilde{Q}_\phi(y \mid x_0) \approx \widetilde{p}(y \mid x_0)$

Training data $(\hat{x}_0, \hat{y}_1) \sim \widetilde{p}(x_0, y_1)$

Non-terminal classifier $(t > 0)$ $\widetilde{Q}_\phi(y_1, y_2 \mid x_t) \approx \widetilde{p}(y_1, y_2 \mid x_t)$

Training data $(\hat{x}_t, \hat{y}_1, \hat{y}_2) \sim \widetilde{p}(x_t, y_1, y_2)$

Classifier training (diffusion)

Full probabilistic model $\widetilde{p}(\{x_t\}_{t=0}^{T})\widetilde{p}(y_1 \mid x_0)\widetilde{p}(y_2 \mid x_0)$

Compositions $\widetilde{p}(x \mid y_1 = i, y_2 = j)$

Mixtures & guidance require $\widetilde{p}(y \mid x_t)$, $\nabla_{x_t} \log \widetilde{p}(y_1, y_2 \mid x_t)$

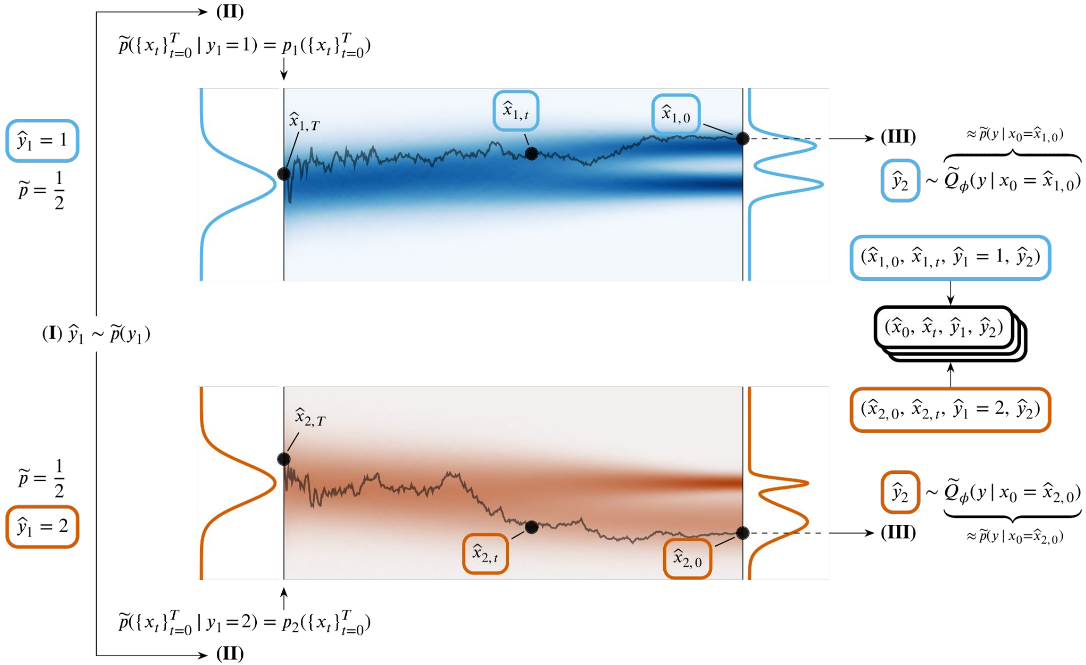
Method: train classifier $\widetilde{Q}_\phi(y_1, y_2 \mid x_t) \approx \widetilde{p}(y_1, y_2 \mid x_t)$

Terminal classifier $(t=0)$ $\widetilde{Q}_\phi(y \mid x_0) \approx \widetilde{p}(y \mid x_0)$

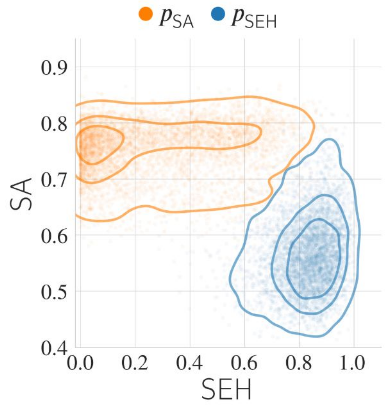
Training data $(\hat{x}_0, \hat{y}_1) \sim \widetilde{p}(x_0, y_1)$

Non-terminal classifier $(t>0)$ $\widetilde{Q}_\phi(y_1, y_2 \mid x_t) \approx \widetilde{p}(y_1, y_2 \mid x_t)$

Training data $(\hat{x}_t, \hat{y}_1, \hat{y}_2) \sim \widetilde{p}(x_t, y_1, y_2)$



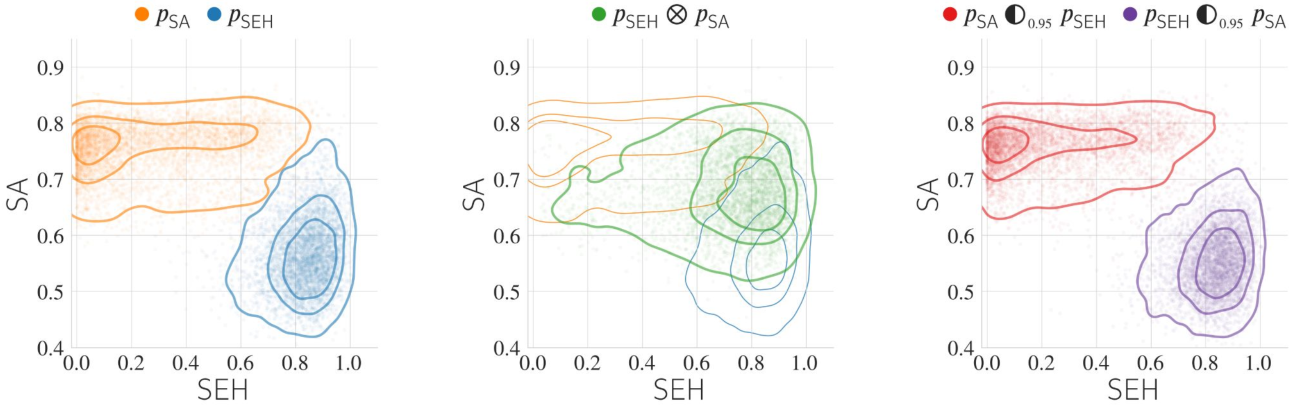
GFlowNets: molecule generation



Rewards

- SEH — learned proxy of a protein binding score (soluble epoxide hydrolase)
- SA — synthetic availability score

GFlowNets: molecule generation



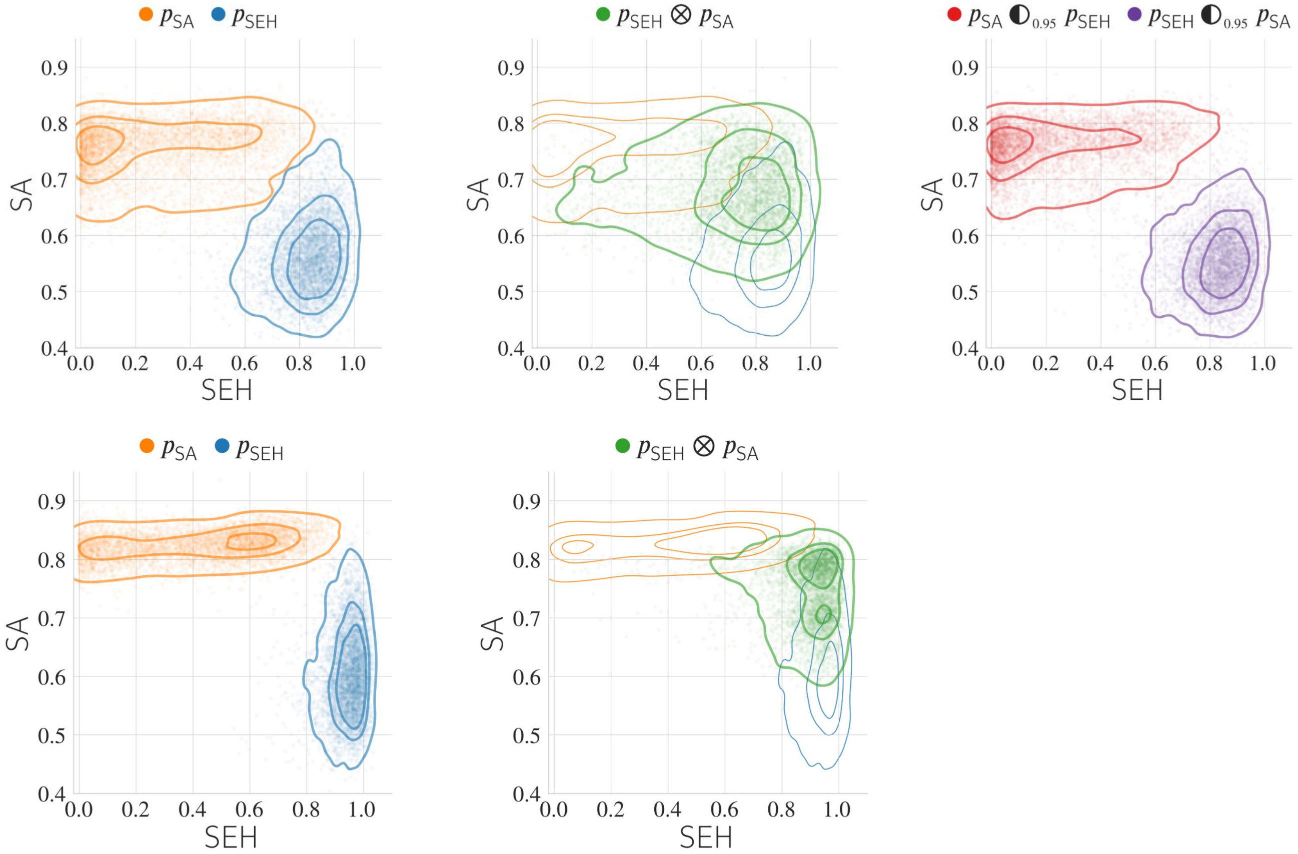
Rewards

- SEH — learned proxy of a protein binding score (soluble epoxide hydrolase)
- SA — synthetic availability score

GFlowNets: molecule generation

$\beta = 32$

$\beta = 96$



Conclusions

MIT CSAIL · Aalto University School of Science · MANCHESTER 1824 The University of Manchester · YaiYai

- New method for composing diffusion models or GFlowNets.
- **Controllable compositions** defined through **observations** of which base models generated a sample.
- **Tractable sampling** from compositions using **classifier guidance on mixtures** of base models.

More results in the paper:

- GFlowNets experiments (2D grid domain, molecular domain)
- Image generation with diffusion models (colored MNIST)
- Parameterized and N-ary composition operations
- Method details and analysis of operations

code on GitHub

https://github.com/timgaripov/compositional-sculpting