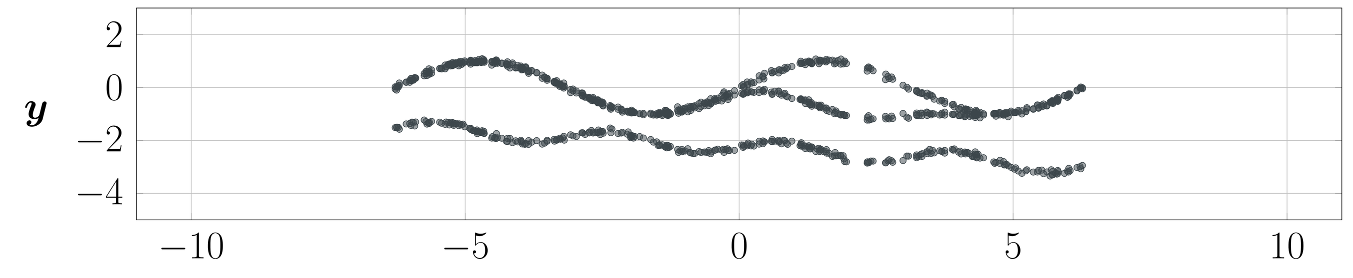
Data Association with Gaussian Processes

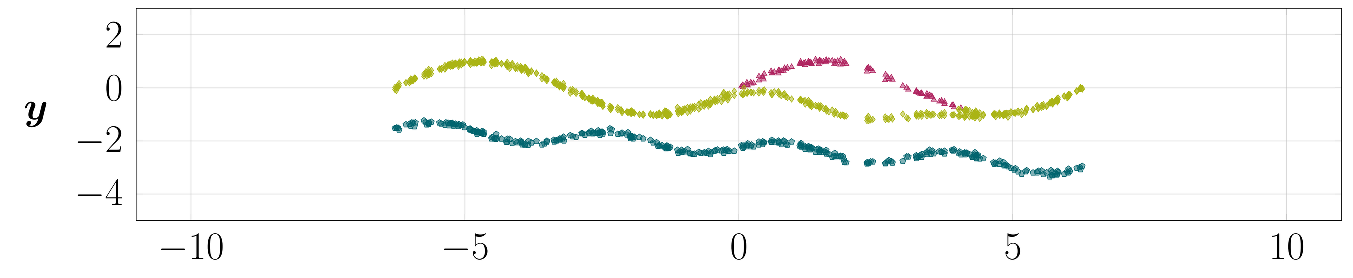
Markus Kaiser [1,2] Clemens Otte [1] Thomas Runkler [1,2] Carl Henrik Ek [3]

[1]Siemens AG [2]Technical University of Munich [3]University of Bristol

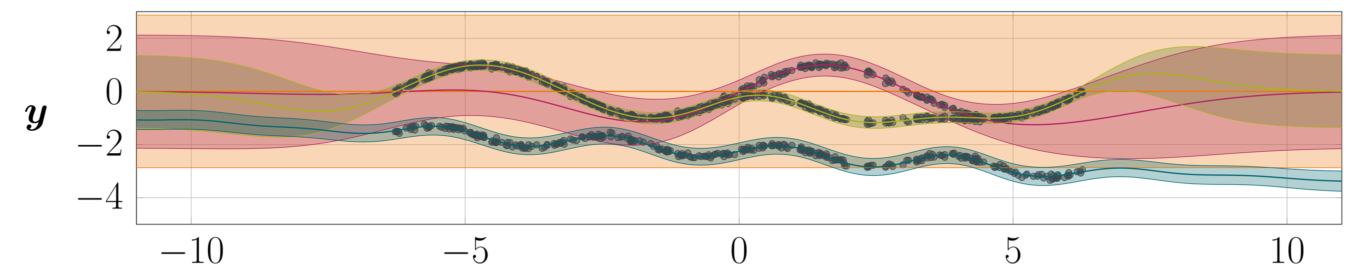
Multimodal data



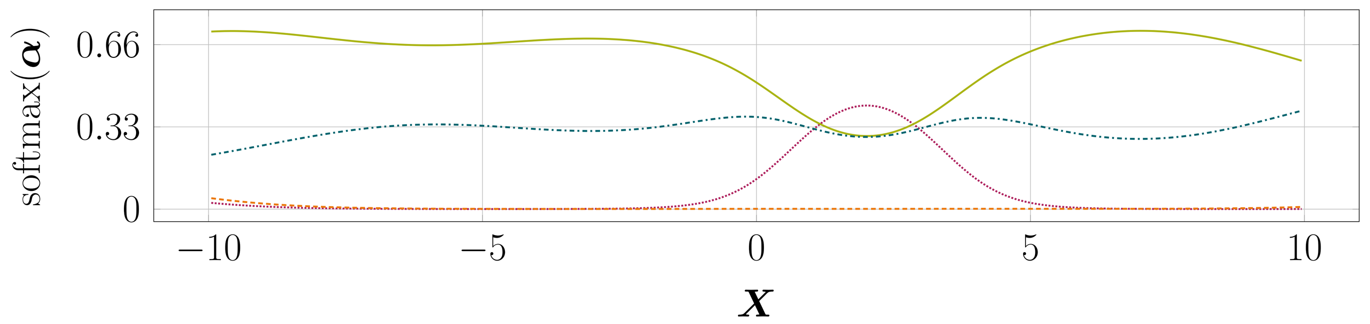
(a) Multimodal dataset



(b) Concrete assignments $\boldsymbol{A}$ learned for the training data



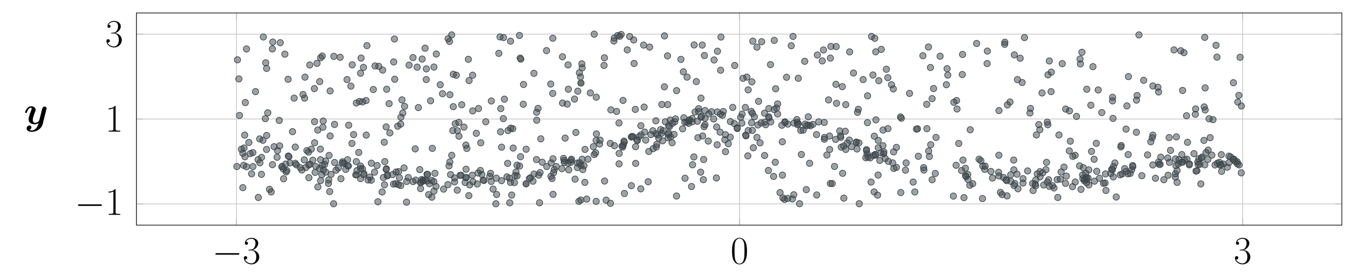
(c) Joint predictions $\boldsymbol{Y}^{(k)}$ of $K = 4$ processes



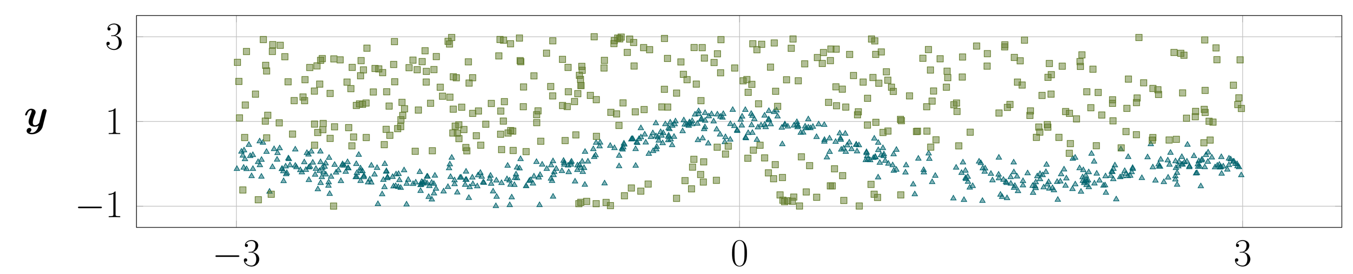
(d) Assignment probabilities $\boldsymbol{\alpha}$ associated with these processes

- DAGP identifies the that three modes are sufficient, one of which is only relevant around $[0, 5]$

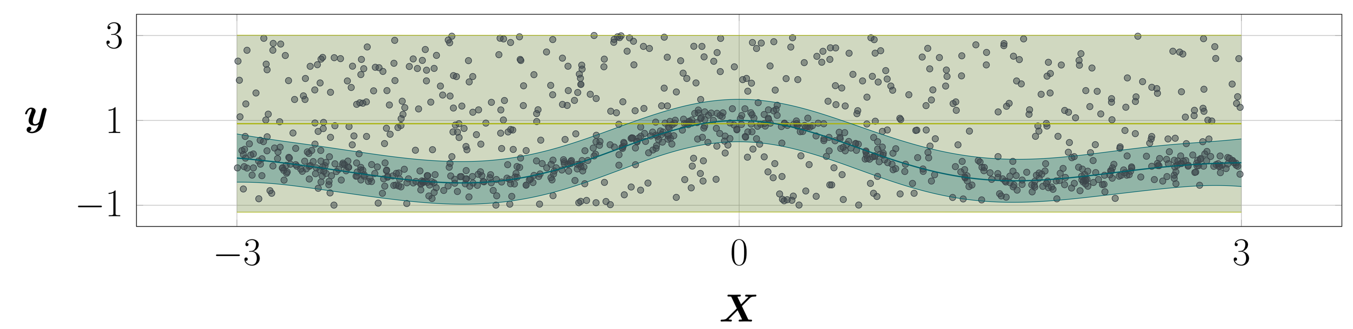
Noise separation



(a) Noise separation problem with 60% outliers



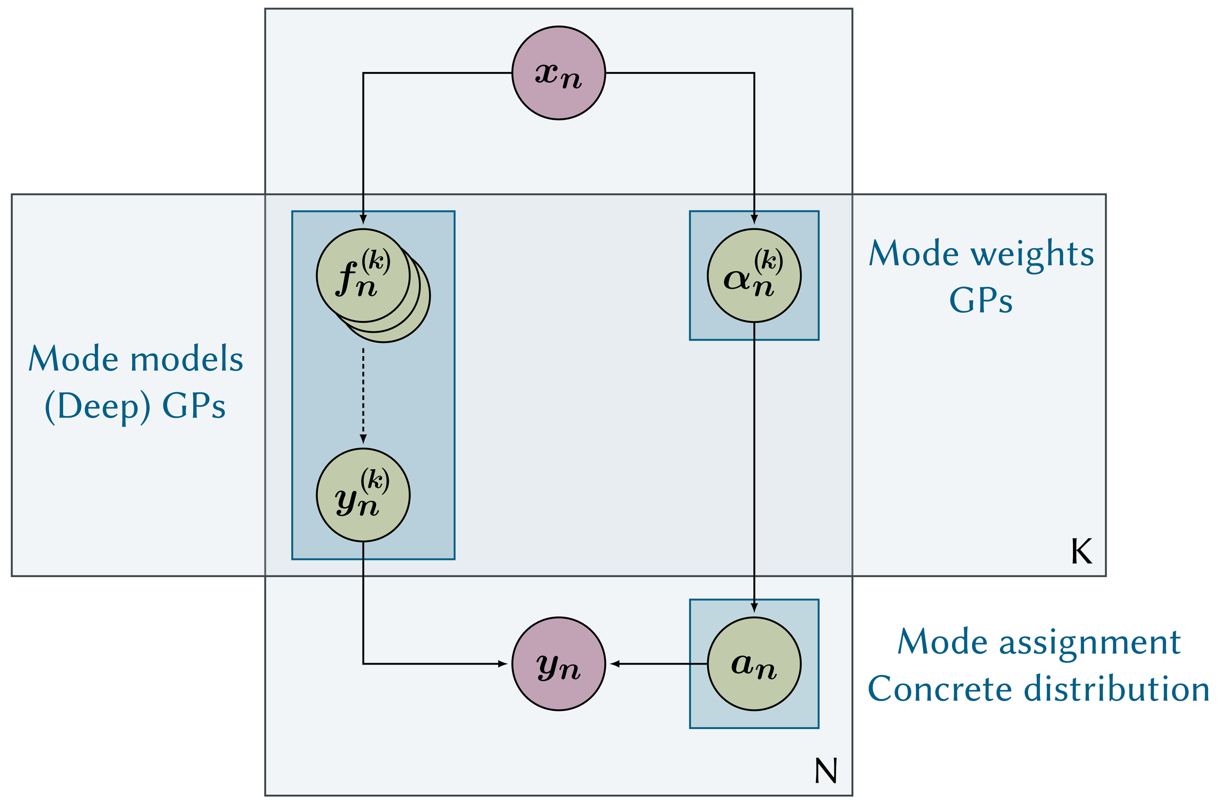
(b) Concrete assignments $\boldsymbol{A}$ learned for the training data



(c) Joint predictions $\boldsymbol{Y}^{(k)}$ of $K = 2$ processes

- DAGP with $K = 2$ processes, one white noise process and one GP with an RBF kernel
- DAGP yields an explicit separation into outliers and signal for the training data

Modelling assumptions of DAGP



- Given a data association problem, DAGP simultaneously learns
  - independent models $(\boldsymbol{F}^{(k)}, \boldsymbol{Y}^{(k)})$ for the K processes
  - concrete assignments $\boldsymbol{A}$ of the training data to these processes
  - a factorization of the input space with respect to the relative importance $\boldsymbol{\alpha}$ of these processes
- DAGP's marginal likelihood combines K regression problems and a classification problem

$$\mathrm{p}(\boldsymbol{Y} \mid \boldsymbol{X}) = \int \mathrm{p}(\boldsymbol{Y} \mid \boldsymbol{F}, \boldsymbol{A})\, \mathrm{p}(\boldsymbol{F} \mid \boldsymbol{X})\, \mathrm{p}(\boldsymbol{A} \mid \boldsymbol{X})\, \mathrm{d}\boldsymbol{A}\, \mathrm{d}\boldsymbol{F}$$

$$\mathrm{p}(\boldsymbol{Y} \mid \boldsymbol{F}, \boldsymbol{A}) = \prod_{n=1}^{N} \prod_{k=1}^{K} \mathcal{N}\left(\boldsymbol{y_n} \,\middle|\, \boldsymbol{f_n^{(k)}}, \left(\sigma^{(k)}\right)^2\right)^{\mathbb{I}(a_n^{(k)}=1)}$$

$$\mathrm{p}(\boldsymbol{A} \mid \boldsymbol{X}) = \int \mathcal{M}(\boldsymbol{A} \mid \mathrm{softmax}(\boldsymbol{\alpha}))\, \mathrm{p}(\boldsymbol{\alpha} \mid \boldsymbol{X})\, \mathrm{d}\boldsymbol{\alpha}$$

Scalable inference

- Learning based on doubly stochastic variational inference by Salimbeni and Deisenroth (2017)
- We add variational inducing variables $\mathrm{q}(\boldsymbol{u} \mid \boldsymbol{Z}) \sim \mathcal{N}(\boldsymbol{u} \mid \boldsymbol{m}, \boldsymbol{S})$ with the factorization

$$\begin{aligned}\mathrm{q}(\boldsymbol{F}, \boldsymbol{\alpha}, \boldsymbol{U}) &= \mathrm{q}\left(\boldsymbol{\alpha}, \left\{\boldsymbol{F}^{(k)}, \boldsymbol{u}^{(k)}, \boldsymbol{u}_{\alpha}^{(k)}\right\}_{k=1}^{K}\right) \\ &= \prod_{k=1}^{K} \prod_{n=1}^{N} \mathrm{p}\left(\boldsymbol{\alpha_n^{(k)}} \,\middle|\, \boldsymbol{u}_{\alpha}^{(k)}, \boldsymbol{x_n}\right) \mathrm{q}\left(\boldsymbol{u}_{\alpha}^{(k)}\right) \prod_{k=1}^{K} \prod_{n=1}^{N} \mathrm{p}\left(\boldsymbol{f_n^{(k)}} \,\middle|\, \boldsymbol{u}^{(k)}, \boldsymbol{x_n}\right) \mathrm{q}\left(\boldsymbol{u}^{(k)}\right).\end{aligned}$$

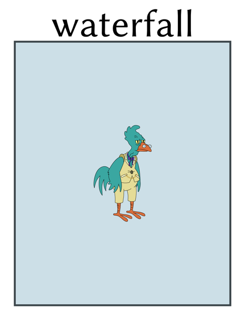
- We use a continuous relaxation of the assignment problem with concrete random variables in $\mathrm{q}(\boldsymbol{A})$
- The joint variational bound is given by

$$\begin{aligned}\mathcal{L}_{\text{DAGP}} &= \mathbb{E}_{\mathrm{q}(\boldsymbol{F}, \boldsymbol{\alpha}, \boldsymbol{U})}\left[\log \frac{\mathrm{p}(\boldsymbol{Y}, \boldsymbol{A}, \boldsymbol{F}, \boldsymbol{\alpha}, \boldsymbol{U} \mid \boldsymbol{X})}{\mathrm{q}(\boldsymbol{F}, \boldsymbol{\alpha}, \boldsymbol{U})}\right] \\ &= \sum_{n=1}^{N} \mathbb{E}_{\mathrm{q}(\boldsymbol{f_n})}[\log \mathrm{p}(\boldsymbol{y_n} \mid \boldsymbol{f_n}, \boldsymbol{a_n})] + \sum_{n=1}^{N} \mathbb{E}_{\mathrm{q}(\boldsymbol{\alpha_n})}[\log \mathrm{p}(\boldsymbol{a_n} \mid \boldsymbol{\alpha_n})] \\ &\quad - \sum_{k=1}^{K} \mathrm{KL}\left(\mathrm{q}\left(\boldsymbol{u}^{(k)}\right) \,\middle\|\, \mathrm{p}\left(\boldsymbol{u}^{(k)} \,\middle|\, \boldsymbol{Z}^{(k)}\right)\right) - \sum_{k=1}^{K} \mathrm{KL}\left(\mathrm{q}\left(\boldsymbol{u}_{\alpha}^{(k)}\right) \,\middle\|\, \mathrm{p}\left(\boldsymbol{u}_{\alpha}^{(k)} \,\middle|\, \boldsymbol{Z}_{\alpha}^{(k)}\right)\right)\end{aligned}$$

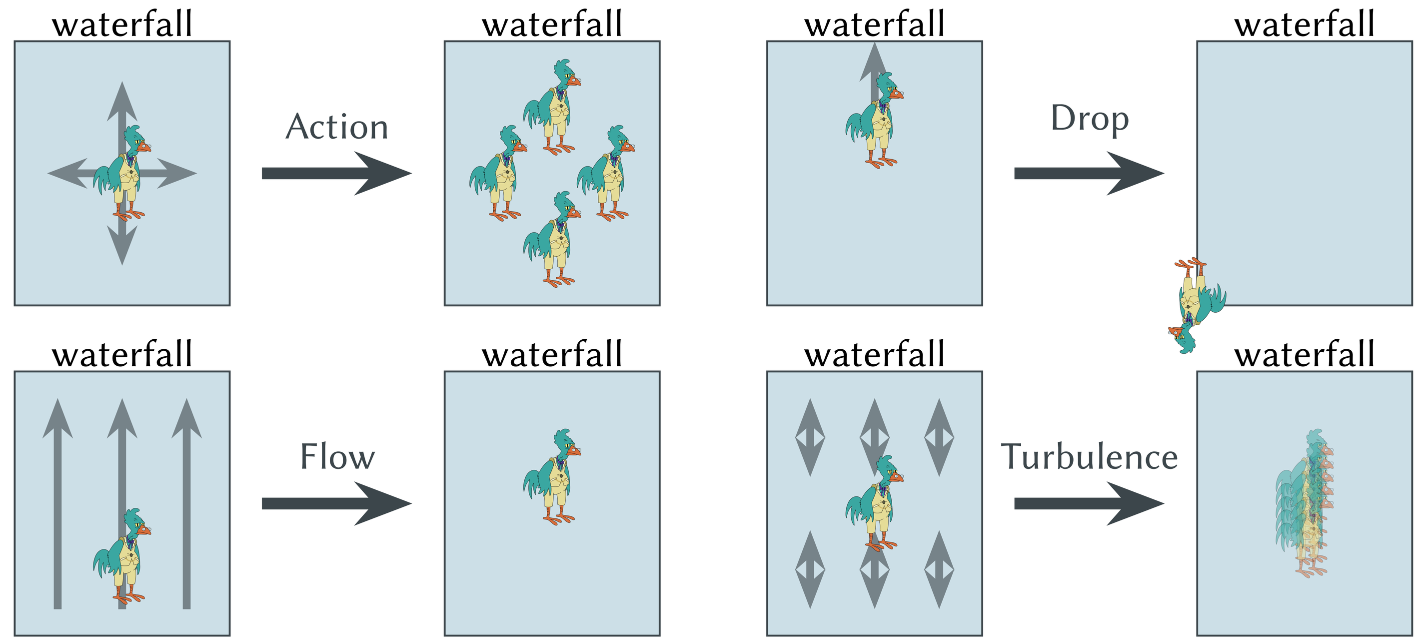
- This bound can be sampled efficiently and factorizes along the data allowing for mini-batches
- The predictive posterior can be efficiently approximated via sampling

$$\begin{aligned}\mathrm{q}(\boldsymbol{f_*} \mid \boldsymbol{x_*}) &= \int \sum_{k=1}^{K} \mathrm{q}\left(a_*^{(k)} \,\middle|\, \boldsymbol{x_*}\right) \mathrm{q}\left(\boldsymbol{f_*^{(k)}} \,\middle|\, \boldsymbol{x_*}\right) \mathrm{d}\boldsymbol{a_*^{(k)}} \\ &\approx \sum_{k=1}^{K} \hat{a}_*^{(k)} \hat{\boldsymbol{f}}_*^{(k)}\end{aligned}$$

Reinforcement Learning

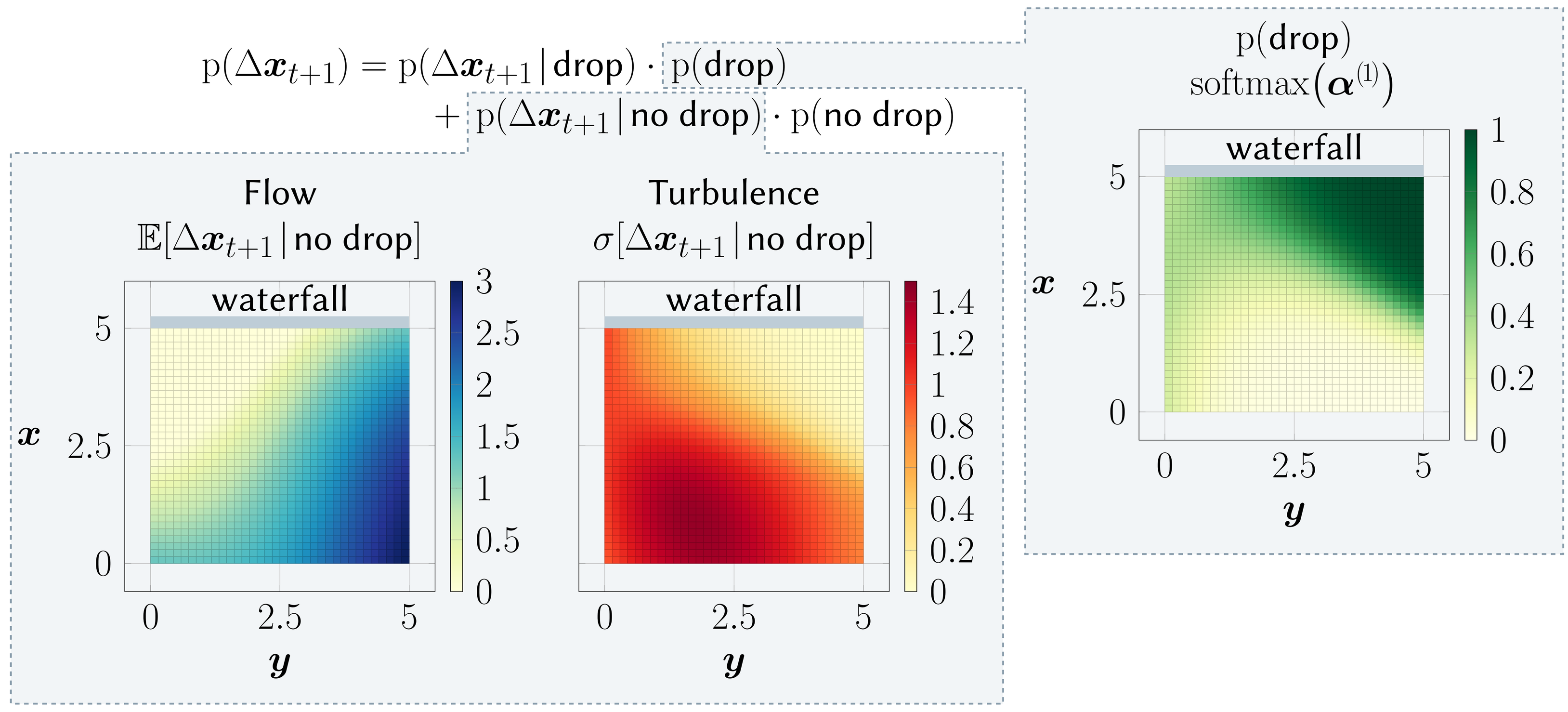


Dynamics	Wet-Chicken agent in a flowing river
Goal	Get close to the waterfall
State	(x, y)-position in $\mathbb{R}^2$
Action	(x, y)-movement in $\mathbb{R}^2$
Challenges	Multimodality and heteroskedasticity



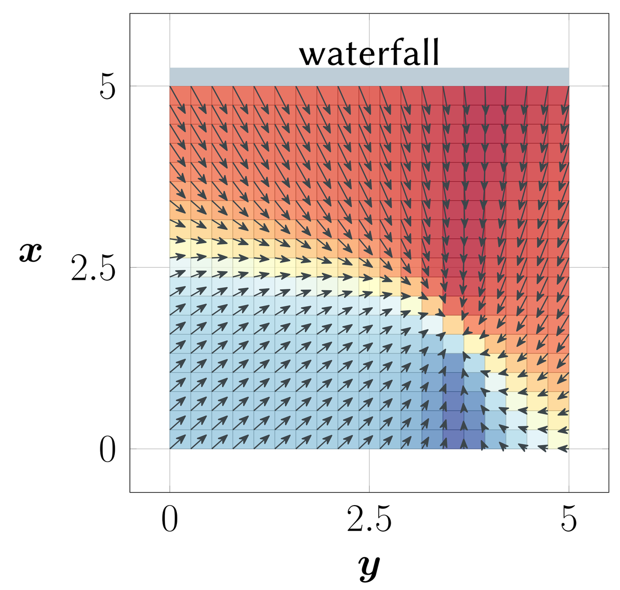
Interpretable transition model

$$\mathrm{p}(\Delta \boldsymbol{x}_{t+1}) = \mathrm{p}(\Delta \boldsymbol{x}_{t+1} \mid \text{drop}) \cdot \mathrm{p}(\text{drop}) + \mathrm{p}(\Delta \boldsymbol{x}_{t+1} \mid \text{no drop}) \cdot \mathrm{p}(\text{no drop})$$

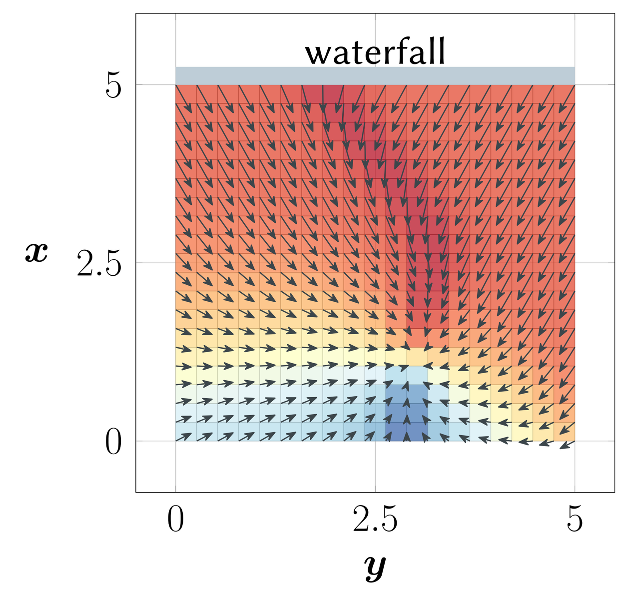


- DAGP's model structure yields interpretable sub-models using only abstract prior knowledge

Conservative policy



(a) Policy using standard reward



(b) Policy using conservative reward

- The sub-models of the DAGP transition model can be used for reward shaping to avoid drops

$$R_{\text{conservative}}(x, y) = R_{\text{original}}(x, y) - 5 \cdot \mathrm{p}(\text{drop} \mid x, y)$$