Distributional Learning: from Methodology to Applications

Xinwei Shen

Seminar for Statistics, ETH Zurich

November 13, 2024

Distributional target

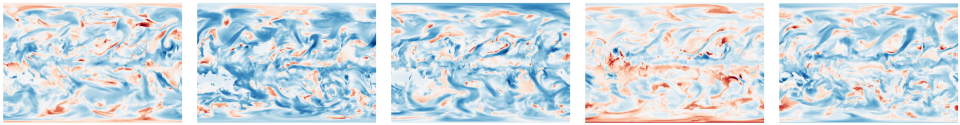
Target: the distribution, rather than merely the mean or median

Distributional target

Target: the distribution, rather than merely the mean or median

- Climate science: precipitation (mean, variation, extremes, spatial structure, etc)

Global precipitation fields on different days

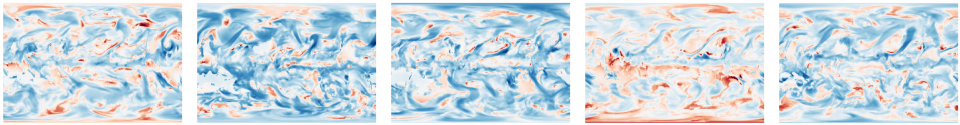


Distributional target

Target: the distribution, rather than merely the mean or median

- Climate science: precipitation (mean, variation, extremes, spatial structure, etc)
- Ecology: species richness (quantiles, extremes)
- ...

Global precipitation fields on different days



Regression

Response $Y \in \mathbb{R}^p$; predictors $X \in \mathbb{R}^d$

Target: $P_{\mathrm{tr}}(y|x)$

Regression

Response $Y \in \mathbb{R}^p$; predictors $X \in \mathbb{R}^d$

$$\textit{Target: } P_{\mathrm{tr}}(y|x)$$

- L_2 or L_1 regression (Legendre, 1806) for conditional mean or median estimation
- Distributional regression via the cdf (Foresi and Peracchi '95; Hothorn et al. '14), pdf (Dunson et al. '07), or quantiles (Koenker et al. '78; Koenker '05; Meinshausen '06) for conditional distribution estimation

Regression

Response $Y \in \mathbb{R}^p$; predictors $X \in \mathbb{R}^d$

Target: $P_{\mathrm{tr}}(y|x)$

- L_2 or L_1 regression (Legendre, 1806) for conditional mean or median estimation
- Distributional regression via the cdf (Foresi and Peracchi '95; Hothorn et al. '14), pdf (Dunson et al. '07), or quantiles (Koenker et al. '78; Koenker '05; Meinshausen '06) for conditional distribution estimation

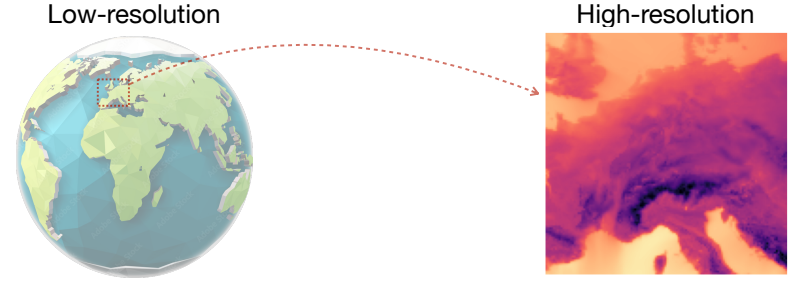
Enough?

High-dimensional response variables

Motivating application: climate downscaling

High-dimensional response variables

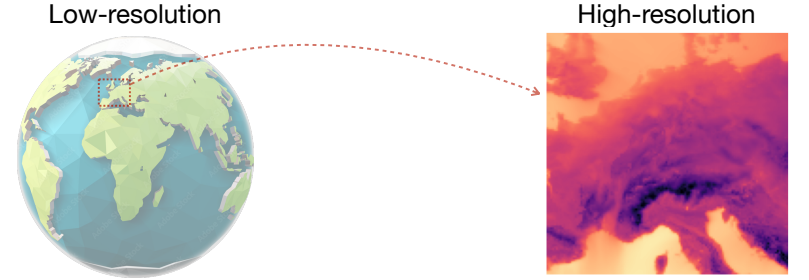
- Physical climate models



Motivating application: climate downscaling

High-dimensional response variables

- Physical climate models



- Statistical downscaling: emulating RCM by estimating $P_{Y|X}$

Distributional learning via generative modeling

- Build a generative model to describe the target distribution:

$$Y = g(X, \varepsilon)$$

where $\varepsilon \sim P_\varepsilon$ pre-defined and map $g : (x, \varepsilon) \mapsto y$ is often parametrized by neural networks.

Distributional learning via generative modeling

- Build a generative model to describe the target distribution:

$$Y = g(X, \varepsilon)$$

  where $\varepsilon \sim P_\varepsilon$ pre-defined and map $g : (x, \varepsilon) \mapsto y$ is often parametrized by neural networks.

- Rationality: change of variables + universal approximation

Distributional learning via generative modeling

- Build a generative model to describe the target distribution:

$$Y = g(X, \varepsilon)$$

  where $\varepsilon \sim P_\varepsilon$ pre-defined and map $g : (x, \varepsilon) \mapsto y$ is often parametrized by neural networks.
- Rationality: change of variables + universal approximation
- Goal: find g such that $g(x, \varepsilon) \sim P_{\mathrm{tr}}(\cdot|x)$ for any x

Distributional learning via generative modeling

- Build a generative model to describe the target distribution:

$$Y = g(X, \varepsilon)$$

  where $\varepsilon \sim P_\varepsilon$ pre-defined and map $g : (x, \varepsilon) \mapsto y$ is often parametrized by neural networks.
- Rationality: change of variables + universal approximation
- Goal: find g such that $g(x, \varepsilon) \sim P_{\mathrm{tr}}(\cdot|x)$ for any x
- Sampling-based inference: a model to sample from $P_{\mathrm{tr}}(y|x)$.

Our distributional learning method: Engression (S. and Meinshausen, '23)

Model class: $\mathcal{M} = \{g(x, \varepsilon)\}$, where ε is a standard Gaussian. Denote $g(x, \varepsilon) \sim P_g(\cdot|x)$.

Our distributional learning method: Engression (S. and Meinshausen, '23)

Model class: $\mathcal{M} = \{g(x, \varepsilon)\}$, where ε is a standard Gaussian. Denote $g(x, \varepsilon) \sim P_g(\cdot|x)$.

Engression: **En**ergy score re**gression**

$$\tilde{g} \in \operatorname*{argmin}_{g \in \mathcal{M}} \mathbb{E}_{(X,Y)\sim P_{\mathrm{tr}}}[-\mathrm{ES}(P_g(y|X), Y)]$$

Our distributional learning method: Engression (S. and Meinshausen, '23)

Model class: $\mathcal{M} = \{g(x, \varepsilon)\}$, where ε is a standard Gaussian. Denote $g(x, \varepsilon) \sim P_g(\cdot|x)$.

Engression: **En**ergy score re**gression**

$$\tilde{g} \in \operatorname*{argmin}_{g \in \mathcal{M}} \mathbb{E}_{(X,Y)\sim P_{\mathrm{tr}}}[-\mathrm{ES}(P_g(y|X), Y)]$$

Energy score (Gneiting and Raftery, '07)

Definition. Given a distribution P and an observation z, the energy score is defined as

$$\mathrm{ES}(P, z) = \frac{1}{2}\mathbb{E}_{(Z,Z')\sim P\otimes P}\|Z - Z'\|_2 - \mathbb{E}_P\|Z - z\|_2.$$

Our distributional learning method: Engression (S. and Meinshausen, '23)

Model class: $\mathcal{M} = \{g(x, \varepsilon)\}$, where ε is a standard Gaussian. Denote $g(x, \varepsilon) \sim P_g(\cdot|x)$.

Engression: **En**ergy score re**gression**

$$\tilde{g} \in \operatorname*{argmin}_{g \in \mathcal{M}} \mathbb{E}_{(X,Y)\sim P_{\mathrm{tr}}}[-\mathrm{ES}(P_g(y|X), Y)]$$

Energy score (Gneiting and Raftery, '07)

Definition. Given a distribution P and an observation z, the energy score is defined as

$$\mathrm{ES}(P, z) = \frac{1}{2}\mathbb{E}_{(Z,Z')\sim P\otimes P}\|Z - Z'\|_2 - \mathbb{E}_P\|Z - z\|_2.$$

Lemma. For any P, we have $\mathbb{E}_{Z\sim P^\star}[\mathrm{ES}(P, Z)] \leq \mathbb{E}_{Z\sim P^\star}[\mathrm{ES}(P^\star, Z)]$, where "$=$" $\Leftrightarrow P = P^\star$.

Our distributional learning method: Engression (S. and Meinshausen, '23)

Model class: $\mathcal{M} = \{g(x, \varepsilon)\}$, where ε is a standard Gaussian. Denote $g(x, \varepsilon) \sim P_g(\cdot|x)$.

Engression: **En**ergy score re**gression**

$$\tilde{g} \in \operatorname*{argmin}_{g \in \mathcal{M}} \mathbb{E}_{(X,Y)\sim P_{\mathrm{tr}}}[-\mathrm{ES}(P_g(y|X), Y)]$$

Energy score (Gneiting and Raftery, '07)

Definition. Given a distribution P and an observation z, the energy score is defined as

$$\mathrm{ES}(P, z) = \frac{1}{2}\mathbb{E}_{(Z,Z')\sim P\otimes P}\|Z - Z'\|_2 - \mathbb{E}_P\|Z - z\|_2.$$

Lemma. For any P, we have $\mathbb{E}_{Z\sim P^\star}[\mathrm{ES}(P, Z)] \le \mathbb{E}_{Z\sim P^\star}[\mathrm{ES}(P^\star, Z)]$, where "$=$" $\Leftrightarrow P = P^\star$.

Corollary. Under correct model specification, we have $\tilde{g}(x, \varepsilon) \sim P_{\mathrm{tr}}(y|x)$, $\forall x \in \mathrm{supp}(P_{\mathrm{tr}}(x))$.

Engression (S. and Meinshausen, '23)

Engression (explicitly):

$$\min_{g \in \mathcal{M}} \mathbb{E}\Big[\|Y - g(X, \varepsilon)\|_2 - \frac{1}{2}\|g(X, \varepsilon) - g(X, \varepsilon')\|_2\Big]$$

Engression (S. and Meinshausen, '23)

Engression (explicitly):

$$\min_{g \in \mathcal{M}} \mathbb{E}\Big[\|Y - g(X, \varepsilon)\|_2 - \frac{1}{2}\|g(X, \varepsilon) - g(X, \varepsilon')\|_2\Big]$$

- Parametrized by neural networks
- Optimized by gradient-based algorithms

Engression (S. and Meinshausen, '23)

Engression (explicitly):

$$\min_{g \in \mathcal{M}} \mathbb{E}\left[\|Y - g(X, \varepsilon)\|_2 - \frac{1}{2}\|g(X, \varepsilon) - g(X, \varepsilon')\|_2\right]$$

- Parametrized by neural networks
- Optimized by gradient-based algorithms

Point estimation by Monte Carlo: for fixed x, draw samples of ε

- Conditional mean estimation: $\hat{\mathbb{E}}_\varepsilon[\tilde{g}(x, \varepsilon)]$
- Conditional α-quantile estimation: $\hat{Q}_\alpha(\tilde{g}(x, \varepsilon))$

Our R and Python packages (`http://github.com/xwshen51/engression`)

R: `install.packages("engression")`

Python: `pip install engression`

Support general data types and tasks:

- X, Y can be multivariate; continuous or categorical
- Estimation for the conditional mean or quantiles
- Sampling from the estimated distribution

Our R and Python packages (http://github.com/xwshen51/engression)

R: `install.packages("engression")`

Python: `pip install engression`

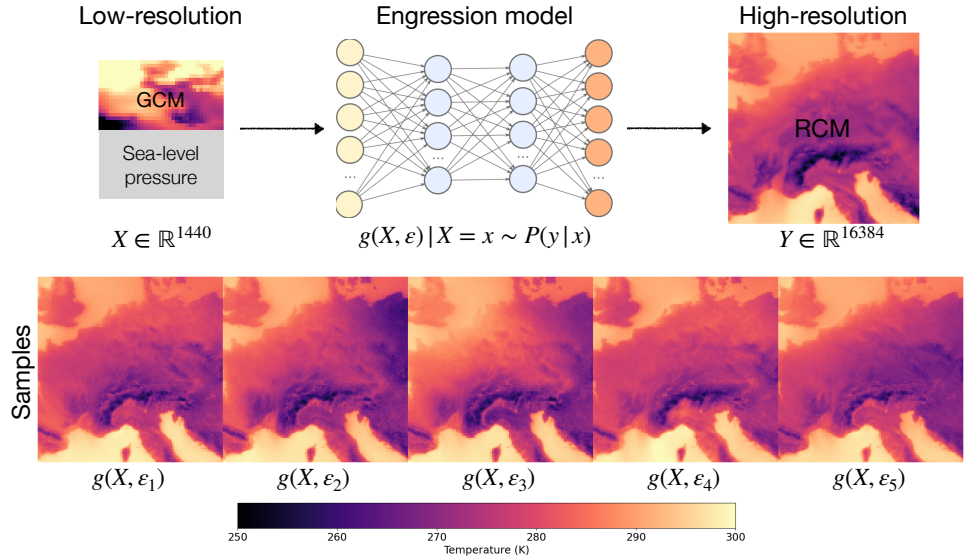
Support general data types and tasks:

- X, Y can be multivariate; continuous or categorical
- Estimation for the conditional mean or quantiles
- Sampling from the estimated distribution

Demo:

```
> library(engression)                                                    ## load engression package
> engressionFit = engression(X, Y)                                       ## fit an engression model
> predict(engressionFit, Xtest, type="mean")                             ## mean prediction
> predict(engressionFit, Xtest, type="quantile", quantiles=c(0.1, 0.5, 0.9)) ## quantile prediction
> predict(engressionFit, Xtest, type="sample", nsample=100)              ## sampling
```

Engression for downscaling (Joint with Maybritt Schillinger, Maxim Samarin, and Nicolai Meinshausen)



Engression as a general distributional learning method

- Estimate the (conditional) distribution

Engression as a general distributional learning method

- Estimate the (conditional) distribution
- Compared to traditional distributional regression (e.g., quantile regression):
  - no quantile crossing
  - expressive capacity of neural networks alleviates limitations of parametric model specifications
  - scalable to (very) high-dimensional X and Y

Engression as a general distributional learning method

- Estimate the (conditional) distribution
- Compared to traditional distributional regression (e.g., quantile regression):
  - no quantile crossing
  - expressive capacity of neural networks alleviates limitations of parametric model specifications
  - scalable to (very) high-dimensional X and Y
- Compared to modern generative models (e.g., diffusion model, GAN):
  - computationally lighter, fewer tuning parameters

Applications of engression to statistical problems

that involve distribution estimation:

- estimation of distributional treatment effects[1]
- distributionally lossless dimension reduction[2]

that require estimating more for identification:

- extrapolation in nonparametric regression[3]
- "under-identified" instrumental variable regression[1]

[1] Holovchak, Saengkyongam, Meinshausen, and S., "Distributional Instrumental Variable Regression," 2024+
[2] S. and Meinshausen, "Distributional Principal Autoencoders," arXiv:2404.13649
[3] S. and Meinshausen, "Engression: Extrapolation through the Lens of Distributional Regression," *JRSSB*, 2024+

Applications of engression to statistical problems

that involve distribution estimation:

- estimation of distributional treatment effects[1] (Part II)
- distributionally lossless dimension reduction[2] (Part III)

that require estimating more for identification:

- extrapolation in nonparametric regression[3] (Part I)
- "under-identified" instrumental variable regression[1] (Part II)

[1] Holovchak, Saengkyongam, Meinshausen, and S., "Distributional Instrumental Variable Regression," 2024+
[2] S. and Meinshausen, "Distributional Principal Autoencoders," arXiv:2404.13649
[3] S. and Meinshausen, "Engression: Extrapolation through the Lens of Distributional Regression," *JRSSB*, 2024+

Application I: Extrapolation in nonparametric regression[1]

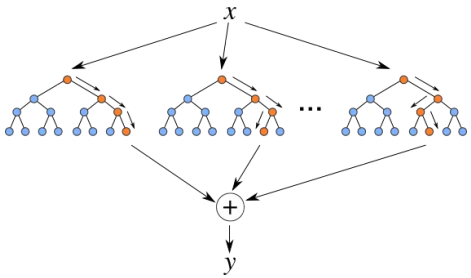
[1] S. and Meinshausen, "Engression: Extrapolation through the Lens of Distributional Regression," *JRSSB*, 2024+

Today's prediction models

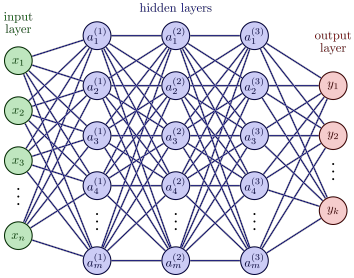
Linear models

$$Y = \beta^\top X + \varepsilon$$

Random Forests, gradient-boosted trees



Neural networks



What could go wrong?

It is common to observe training data within a bounded support and encounter test data outside the training support.

- Biodiversity: predicting how species respond to climate change
- Counterfactual prediction: covariate shifts from the treatment to control groups
- ...

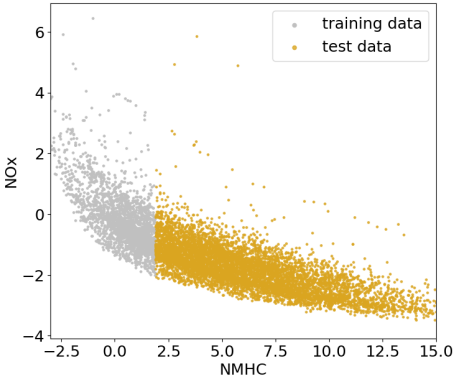
What could go wrong?

It is common to observe training data within a bounded support and encounter test data outside the training support.

- Biodiversity: predicting how species respond to climate change
- Counterfactual prediction: covariate shifts from the treatment to control groups
- ...

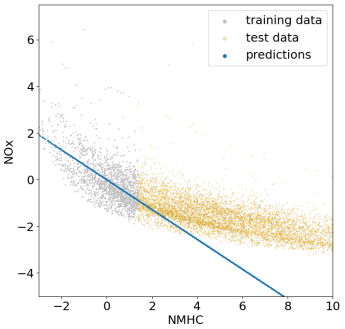
Extrapolation is a fundamental challenge for nonparametric regression.

Air quality data example



Measurements of two pollutants: Total Nitrogen Oxides (NOx) and non-methane hydrocarbons (NMHC) concentration.

Challenge of nonlinear extrapolation



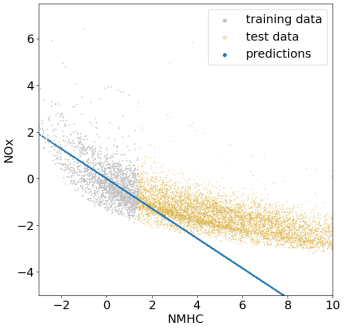
Linear regression

Random Forests

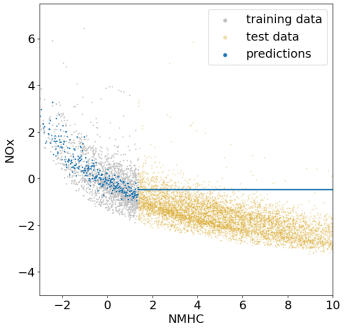
Neural network regression[1]

[1]Predictions from different random initializations and NN architectures with 3, 5, 7, or 9 layers

Challenge of nonlinear extrapolation



Linear regression

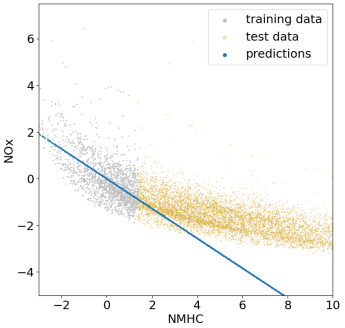


Random Forests

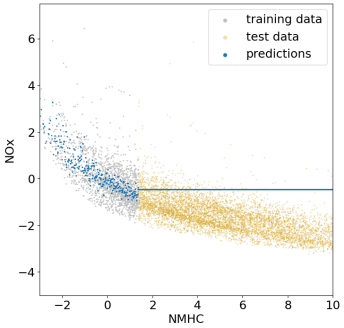
Neural network regression[1]

[1] Predictions from different random initializations and NN architectures with 3, 5, 7, or 9 layers

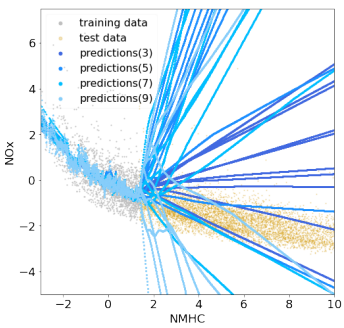
Challenge of nonlinear extrapolation



Linear regression



Random Forests

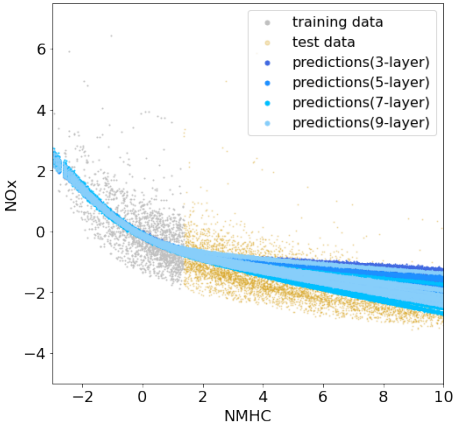


Neural network regression[1]

[1]Predictions from different random initializations and NN architectures with 3, 5, 7, or 9 layers

Engression makes a difference

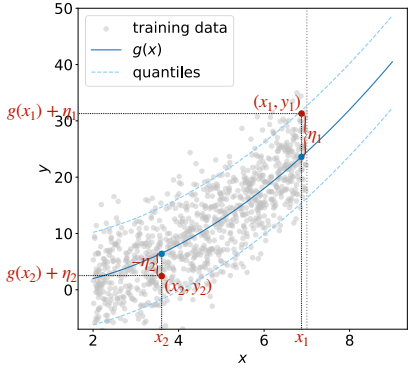
The reliability of engression does not break down immediately at the support boundary.



Results of engression with 3, 5, 7, or 9 layers and random initializations.

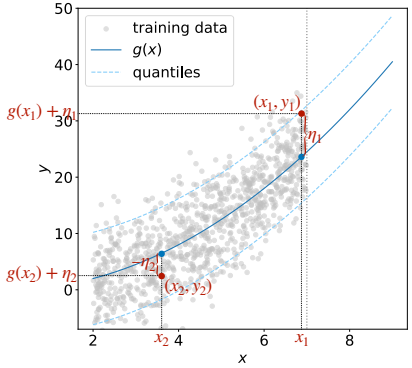
Additive noise models (ANMs)

Post-ANM: $Y = g(X) + \eta$

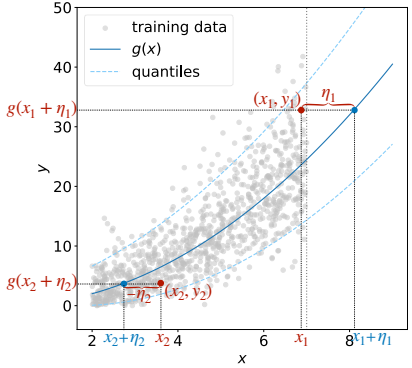


Additive noise models (ANMs)

Post-ANM: $Y = g(X) + \eta$

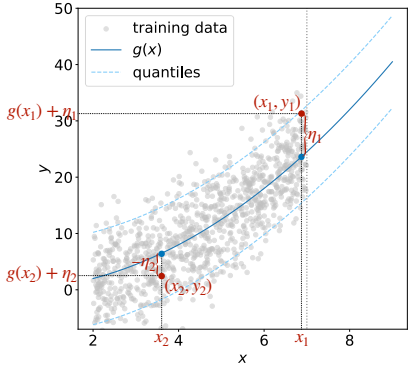


Pre-ANM: $Y = g(X + \eta)$

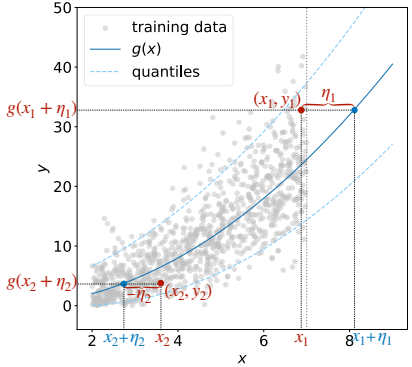


Additive noise models (ANMs)

Post-ANM: $Y = g(X) + \eta$



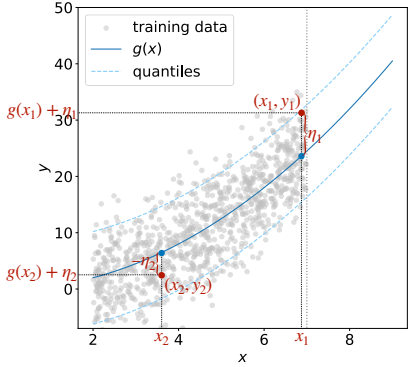
Pre-ANM: $Y = g(X + \eta)$



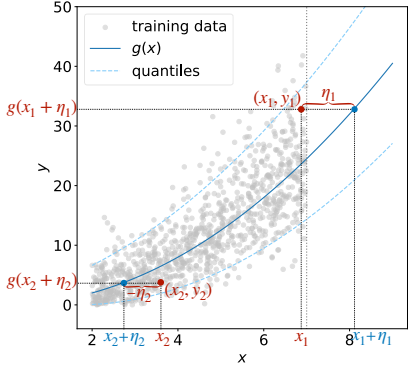
All models are wrong, but can one of them be useful in terms of extrapolation?

Additive noise models (ANMs)

Post-ANM: $Y = g(X) + \eta$

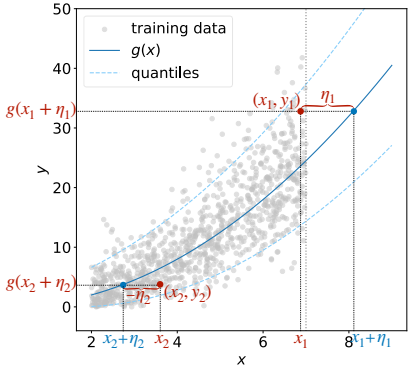


Pre-ANM: $Y = g(X + \eta)$

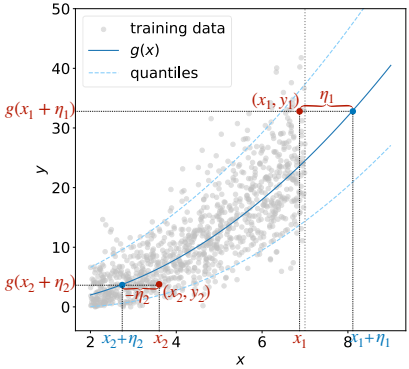


Pre-additive noises reveal some information about the true function outside the support.

Pre-ANM: $Y = g(X + \eta)$



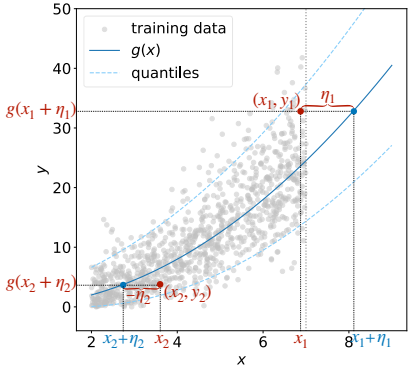
Pre-ANM: $Y = g(X + \eta)$



To capture the information from the pre-additive noise

Distributional learning

Pre-ANM: $Y = g(X + \eta)$



To capture the information from the pre-additive noise, one needs to fit the full conditional distribution of Y given X.

two ingredients for extrapolation

a distributional learning method

pre-ANMs

Engression has the two ingredients for extrapolation

- ✓ Engression is **a distributional learning method**.
- ✓ Engression model $\mathcal{M} = \{g(x, \varepsilon)\}$ contains **pre-ANMs** $\{g(W^\top x + h(\varepsilon)) : g \in \mathcal{G}, h \in \mathcal{H}\}$, where $h(\varepsilon)$ represents the pre-additive noise; g, h, and W are to be learned.

Setup:

- True model $Y = g^\star(X + \eta)$; pre-ANM class $\mathcal{M} = \{g(x + h(\varepsilon)) : g \in \mathcal{G}, h \in \mathcal{H}\}$; $\mathcal{G}$ strictly monotone;
- (For simplicity) symmetric noise $\eta \in [-\eta_{\max}, \eta_{\max}]$; training support $(-\infty, x_{\max}]$.

Regression fails to extrapolate

Setup:

- True model $Y = g^\star(X + \eta)$; pre-ANM class $\mathcal{M} = \{g(x + h(\varepsilon)) : g \in \mathcal{G}, h \in \mathcal{H}\}$; $\mathcal{G}$ strictly monotone;
- (For simplicity) symmetric noise $\eta \in [-\eta_{\max}, \eta_{\max}]$; training support $(-\infty, x_{\max}]$.

Proposition (S. and Meinshausen, '23)

Let $\mathcal{F}_{L_1} := \operatorname{argmin}_{g \in \mathcal{G}} \mathbb{E}_{P_{\text{tr}}} |Y - g(X)|$. For any $x > x_{\max}$, we have

$$\sup_{g \in \mathcal{F}_{L_1}} |g(x) - g^\star(x)| = \infty.$$

Engression can extrapolate up to a certain point

Setup:

- True model $Y = g^\star(X + \eta)$; pre-ANM class $\mathcal{M} = \{g(x + h(\varepsilon)) : g \in \mathcal{G}, h \in \mathcal{H}\}$; $\mathcal{G}$ strictly monotone;
- (For simplicity) symmetric noise $\eta \in [-\eta_{\max}, \eta_{\max}]$; training support $(-\infty, x_{\max}]$.

Theorem (S. and Meinshausen, '23)

We have $\tilde{g}(x) = g^\star(x)$ for all $x \le x_{\max} + \eta_{\max}$, and $\tilde{h}(\varepsilon) \overset{d}{=} \eta$.

- Population engression $(\tilde{g}, \tilde{h})$ recovers the true model beyond the training support.

Engression can extrapolate up to a certain point

Setup:

- True model $Y = g^\star(X + \eta)$; pre-ANM class $\mathcal{M} = \{g(x + h(\varepsilon)) : g \in \mathcal{G}, h \in \mathcal{H}\}$; $\mathcal{G}$ strictly monotone;
- (For simplicity) symmetric noise $\eta \in [-\eta_{\max}, \eta_{\max}]$; training support $(-\infty, x_{\max}]$.

Theorem (S. and Meinshausen, '23)

We have $\tilde{g}(x) = g^\star(x)$ for all $x \le x_{\max} + \eta_{\max}$, and $\tilde{h}(\varepsilon) \overset{d}{=} \eta$.

- Population engression $(\tilde{g}, \tilde{h})$ recovers the true model beyond the training support.
- Blessing of noise: the more (pre-additive) noise there is, the farther one can extrapolate.

Relax the assumptions?

"truth $Y = g^\star(X + \eta)$; pre-ANM class $\mathcal{M} = \{g(x + h(\varepsilon)) : g \in \mathcal{G}, h \in \mathcal{H}\}$; $\mathcal{G}$ monotone"?

- Model $Y = g^\star(X + \eta) + \xi$ to allow both pre and post-additive noises
- Monotone $g^\star$ only around the support boundary.

Relax the assumptions?

"truth $Y = g^\star(X + \eta)$; pre-ANM class $\mathcal{M} = \{g(x + h(\varepsilon)) : g \in \mathcal{G}, h \in \mathcal{H}\}$; $\mathcal{G}$ monotone"?

- Model $Y = g^\star(X + \eta) + \xi$ to allow both pre and post-additive noises
- Monotone $g^\star$ only around the support boundary.

For conditional distribution estimation, engression is rather general.

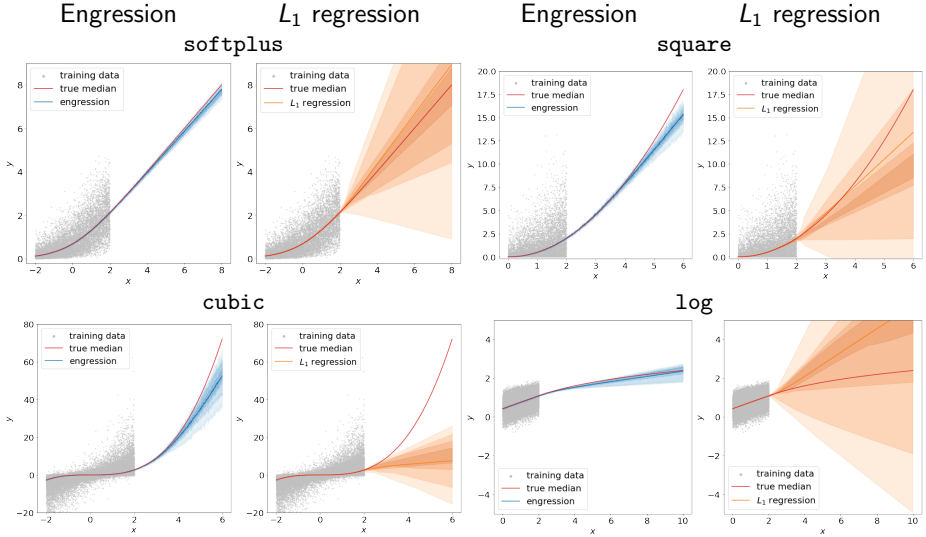
In practice, engression uses general models $\{g(x, \varepsilon)\}$.

Simulation settings

Table: $Y = g^\star(X + \eta)$, $x_{\max} = 2$, $\eta_{\max} \approx 2$

Name	$g^\star(\cdot)$	X	η
`softplus`	$g^\star(x) = \log(1 + e^x)$	Unif$[-2, 2]$	$\mathcal{N}(0, 1)$
`square`	$g^\star(x) = (x_+)^2/2$	Unif$[0, 2]$	$\mathcal{N}(0, 1)$
`cubic`	$g^\star(x) = x^3/3$	Unif$[-2, 2]$	$\mathcal{N}(0, 1.1^2)$
`log`	$g^\star(x) = \begin{cases} \frac{x-2}{3} + \log(3) & x \le 2 \\ \log(x) & x > 2 \end{cases}$	Unif$[0, 2]$	$\mathcal{N}(0, 1)$

Conditional median estimation



Large-scale real-data experiments for univariate prediction

590 data configurations:

- *Real data sets* from various application domains
- *Pairwise prediction* for all variables
- *Split the training and test data* at the 0.3–0.7 quantiles of the predictor

Large-scale real-data experiments for univariate prediction

590 data configurations:

- *Real data sets* from various application domains
- *Pairwise prediction* for all variables
- *Split the training and test data* at the 0.3–0.7 quantiles of the predictor

18 hyperparameter settings of neural network architectures and optimization

In total: 590*18=10'620 models for each method

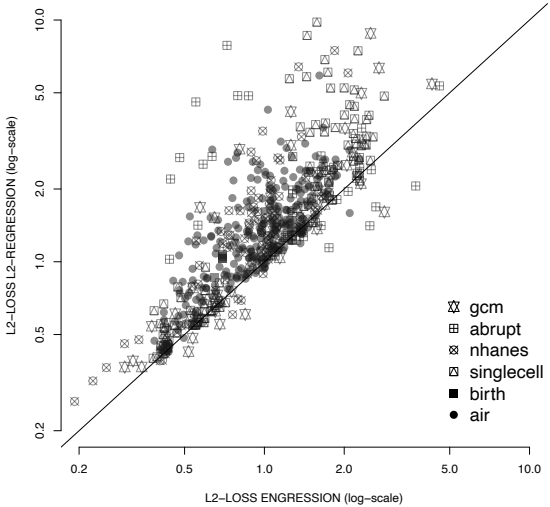
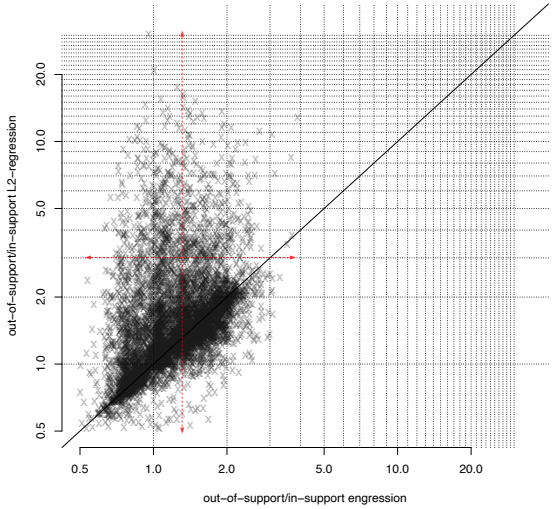
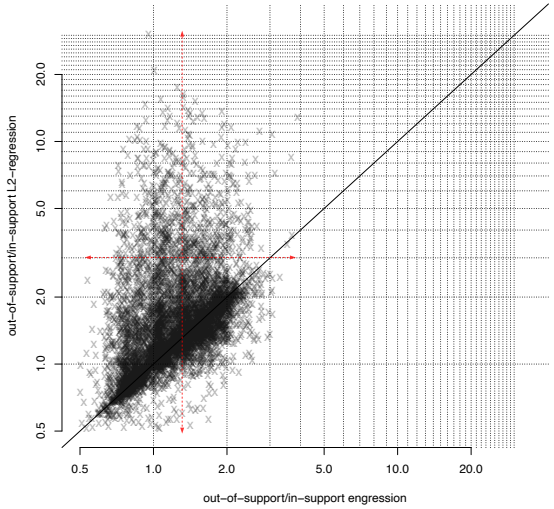


Figure: Out-of-support losses (in log-scale) of engression and regression for various data configurations, averaging over all hyperparameter settings.

The ratio (in log-scale) between out-of-support and in-support L_2 losses of engression and regression for all hyperparameter settings.

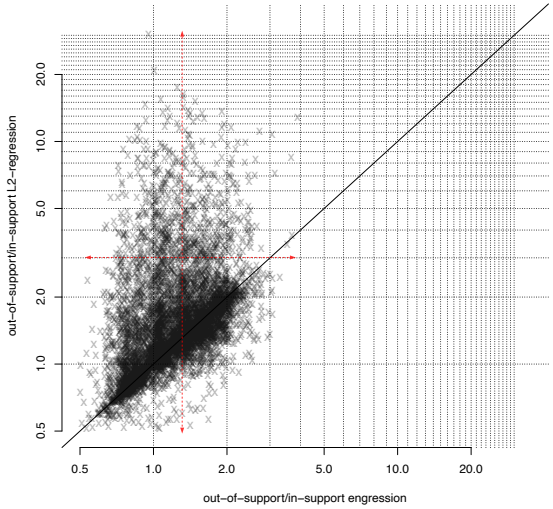


The ratio (in log-scale) between out-of-support and in-support L_2 losses of engression and regression for all hyperparameter settings.



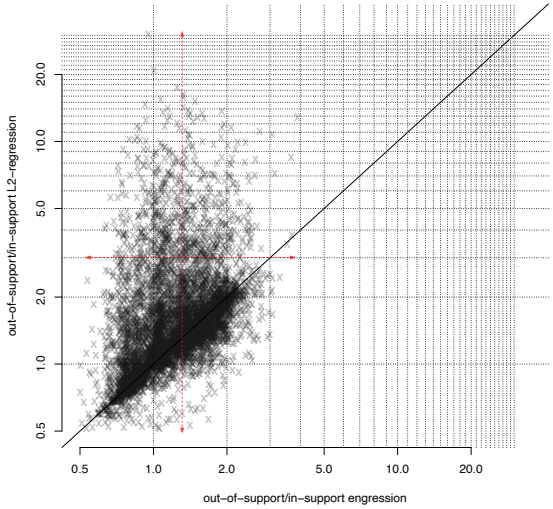
- Engression has comparable out-of-support and in-support performance.

The ratio (in log-scale) between out-of-support and in-support L_2 losses of engression and regression for all hyperparameter settings.



- Engression has comparable out-of-support and in-support performance.
- Regression degrades drastically out-of-support.

The ratio (in log-scale) between out-of-support and in-support L_2 losses of engression and regression for all hyperparameter settings.



- Engression has comparable out-of-support and in-support performance.
- Regression degrades drastically out-of-support.
- Engression is much more robust to the choice of hyperparameters than NN regression.

Prediction intervals

Proposition (S. and Meinshausen, '23)

For $\alpha \in [0,1]$, it holds for all $x \leq x_{\max} + \eta_{\max} - Q_\alpha(\eta)$ that $\tilde{q}_\alpha(x) = q^\star_\alpha(x)$, i.e.,

$$\mathbb{P}\left(Y \leq \tilde{q}_{1-\alpha}(X) \mid X = x\right) = 1 - \alpha.$$

$\Rightarrow$ prediction intervals with conditional coverage guarantee outside the support (population-level)

Prediction intervals

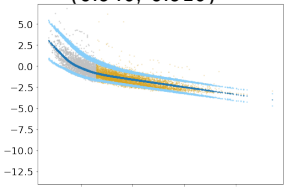
Proposition (S. and Meinshausen, '23)

For $\alpha \in [0,1]$, it holds for all $x \leq x_{\text{max}} + \eta_{\text{max}} - Q_\alpha(\eta)$ that $\tilde{q}_\alpha(x) = q^\star_\alpha(x)$, i.e.,

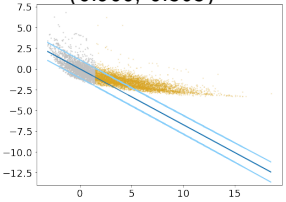
$$\mathbb{P}\left(Y \leq \tilde{q}_{1-\alpha}(X) \mid X = x\right) = 1 - \alpha.$$

$\Rightarrow$ prediction intervals with conditional coverage guarantee outside the support (population-level)

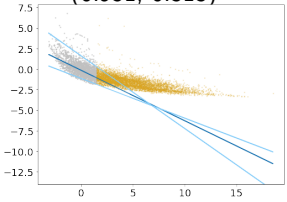
Engression
(0.946, 0.916)



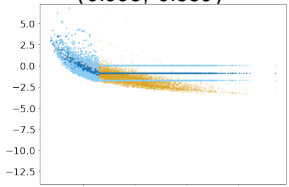
Linear regression
(0.900, 0.363)



Linear quantile regression
(0.951, 0.313)



Quantile regression forests
(0.995, 0.559)



Summary I

Engression + pre-additive noise model $\Rightarrow$ extrapolation

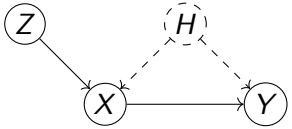
Application II: Causal Effect Estimation[1]

[1] Holovchak, Saengkyongam, Meinshausen, and S., "Distributional Instrumental Variable Regression," 2024+

Instrumental variable model

Treatment X, outcome Y, instrumental variable Z.

$$X \leftarrow g(Z, \eta_X)$$
$$Y \leftarrow f(X, \eta_Y)$$

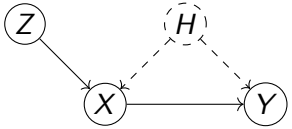


where f, g can be nonlinear, and η_X and η_Y are correlated due to latent confounder H.

Instrumental variable model

Treatment X, outcome Y, instrumental variable Z.

$$X \leftarrow g(Z, \eta_X)$$
$$Y \leftarrow f(X, \eta_Y)$$



where f, g can be nonlinear, and η_X and η_Y are correlated due to latent confounder H.

Estimand: do-interventional distribution $P(Y|do(X := x))$

Identifiability of the estimand

Theorem

Assume for all $z \in \mathrm{supp}(Z)$, $g(z, \cdot)$ is strictly monotone, and for all $x \in \mathrm{supp}(X)$, $\mathrm{supp}(\eta_X | X = x) = \mathrm{supp}(\eta_X)$. Then, for all $x \in \mathrm{supp}(X)$, the interventional distribution $P(Y|do(X := x))$ is uniquely determined from the observed data distribution $P_{\mathrm{tr}}(x, y|z)$.

Identifiability of the estimand

Theorem

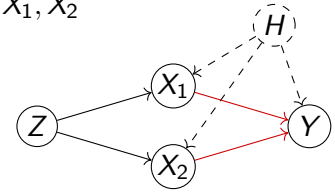
Assume for all $z \in \mathrm{supp}(Z)$, $g(z, \cdot)$ is strictly monotone, and for all $x \in \mathrm{supp}(X)$, $\mathrm{supp}(\eta_X | X = x) = \mathrm{supp}(\eta_X)$. Then, for all $x \in \mathrm{supp}(X)$, the interventional distribution $P(Y|do(X := x))$ is uniquely determined from the observed data distribution $P_{\mathrm{tr}}(x, y|z)$.

Estimate $P_{\mathrm{tr}}(x, y|z) \overset{\text{sufficient}}{\Rightarrow}$ identify $P(Y|do(X := x))$

Under-identified case: the full distribution $P_{\mathrm{tr}}(x, y|z)$ is also "necessary" for identification

- One binary IV $Z \in \{0, 1\}$, two continuous treatments X_1, X_2

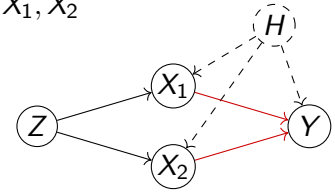
$$
\begin{aligned}
X_1 &= g_1(Z, \eta_1) \\
X_2 &= g_2(Z, \eta_2) \\
Y &= \beta_1 X_1 + \beta_2 X_2 + \eta_Y
\end{aligned}
$$



Under-identified case: the full distribution $P_{\mathrm{tr}}(x, y|z)$ is also "necessary" for identification

- One binary IV $Z \in \{0, 1\}$, two continuous treatments X_1, X_2

$$
\begin{aligned}
X_1 &= g_1(Z, \eta_1) \\
X_2 &= g_2(Z, \eta_2) \\
Y &= \beta_1 X_1 + \beta_2 X_2 + \eta_Y
\end{aligned}
$$

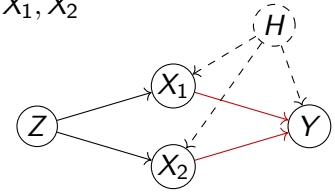


- Two-stage least-squares would **fail** as $\mathbb{E}[X_1|Z]$ and $\mathbb{E}[X_2|Z]$ are collinear.

Under-identified case: the full distribution $P_{\mathrm{tr}}(x, y|z)$ is also "necessary" for identification

- One binary IV $Z \in \{0, 1\}$, two continuous treatments X_1, X_2

$$
\begin{aligned}
X_1 &= g_1(Z, \eta_1) \\
X_2 &= g_2(Z, \eta_2) \\
Y &= \beta_1 X_1 + \beta_2 X_2 + \eta_Y
\end{aligned}
$$



- Two-stage least-squares would **fail** as $\mathbb{E}[X_1|Z]$ and $\mathbb{E}[X_2|Z]$ are collinear.
- Distributional identifiability holds:

Theorem

Assume $(X_i|Z = 0) \overset{d}{\neq} (c + X_i|Z = 1)$, for any constant c, for $i = 1, 2$. Then β_1 and β_2 are uniquely determined from $P_{\mathrm{tr}}(x_1, x_2, y|z)$.

Distributional instrumental variable (DIV) method

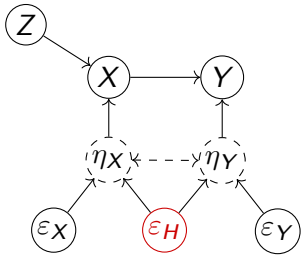
- Joint generative model:

$$\left.\begin{aligned}\eta_X &= h_X(\varepsilon_X, \varepsilon_H)\\ \eta_Y &= h_Y(\varepsilon_Y, \varepsilon_H)\end{aligned}\right\} \text{confounded noises}$$

$$X = g(Z, \eta_X)$$

$$Y = f(X, \eta_Y)$$

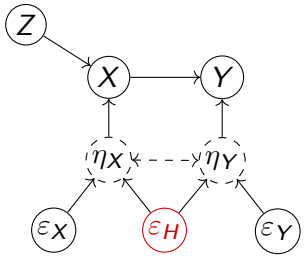
where $\varepsilon_X, \varepsilon_Y, \varepsilon_H$ are independent standard Gaussians.



Distributional instrumental variable (DIV) method

- Joint generative model:

$$\left.\begin{aligned}\eta_X &= h_X(\varepsilon_X, \varepsilon_H)\\ \eta_Y &= h_Y(\varepsilon_Y, \varepsilon_H)\end{aligned}\right\} \text{confounded noises}$$
$$X = g(Z, \eta_X)$$
$$Y = f(X, \eta_Y)$$



where $\varepsilon_X, \varepsilon_Y, \varepsilon_H$ are independent standard Gaussians.

- DIV solution (engression applied to $(X, Y)|Z$):

$$\underset{f,g,h_X,h_Y}{\operatorname{argmin}} \mathbb{E}\left[\|(X,Y) - (\hat{X},\hat{Y})\| - \frac{1}{2}\|(\hat{X},\hat{Y}) - (\hat{X}',\hat{Y}')\|\right],$$

where

$$\hat{X} := g(Z, h_X(\varepsilon_X, \varepsilon_H)) \qquad \hat{Y} := f(\hat{X}, h_Y(\varepsilon_Y, \varepsilon_H))$$
$$\hat{X}' := g(Z, h_X(\varepsilon'_X, \varepsilon'_H)) \qquad \hat{Y}' := f(\hat{X}', h_Y(\varepsilon'_Y, \varepsilon'_H))$$

Illustrative example of DIV

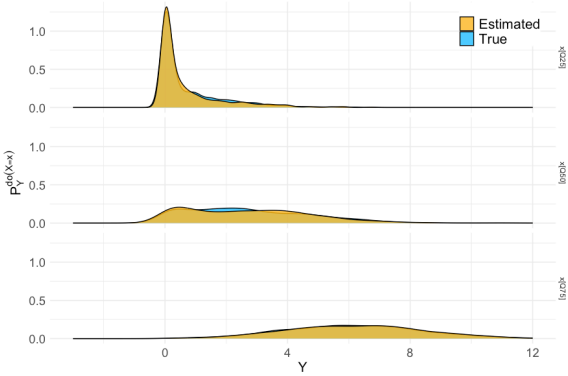
DIV solution f^*, h_Y^* enables sampling from the interventional distribution:

$$f^*(x, h_Y^*(\varepsilon_Y, \varepsilon_H)) \sim P(Y|do(X := x)), \quad \forall x$$

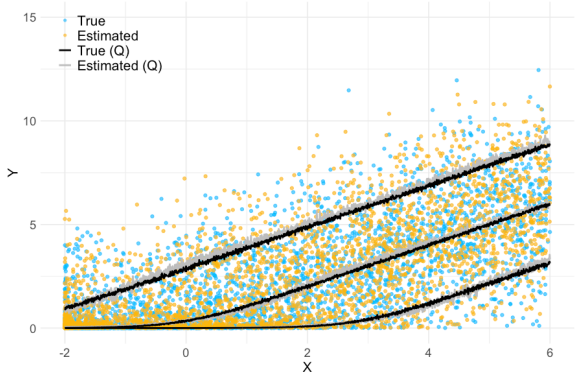
Illustrative example of DIV

DIV solution f^*, h_Y^* enables sampling from the interventional distribution:

$$f^*(x, h_Y^*(\varepsilon_Y, \varepsilon_H)) \sim P(Y|do(X := x)), \quad \forall x$$



Histograms of $P(Y|do(X := x))$ for different x



Samples from $P(Y|do(X := x))$ and quantile treatment effects

Summary II

- Engression + instrumental variable $\Rightarrow$ distributional causal effect estimation
- More identification compared to 2SLS

Application III: Distributionally lossless dimension reduction[1]

[1] S. and Meinshausen, "Distributional Principal Autoencoders," arXiv:2404.13649

Dimension reduction

- Data $X \in \mathbb{R}^p$
- Dimension reduction: *encoder* $e(.) : \mathbb{R}^p \to \mathbb{R}^k$ where $k < p$.

Dimension reduction

- Data $X \in \mathbb{R}^p$
- Dimension reduction: *encoder* $e(.) : \mathbb{R}^p \to \mathbb{R}^k$ where $k < p$.
- Data reconstruction: *decoder* $d(.) : \mathbb{R}^k \to \mathbb{R}^p$.

Dimension reduction

- Data $X \in \mathbb{R}^p$
- Dimension reduction: *encoder* $e(.) : \mathbb{R}^p \to \mathbb{R}^k$ where $k < p$.
- Data reconstruction: *decoder* $d(.) : \mathbb{R}^k \to \mathbb{R}^p$.
- Common criterion: minimising the mean squared reconstruction loss

$$\min \mathbb{E}\left[\|X - d(e(X))\|^2\right]$$

Dimension reduction

- Data $X \in \mathbb{R}^p$
- Dimension reduction: *encoder* $e(.) : \mathbb{R}^p \to \mathbb{R}^k$ where $k < p$.
- Data reconstruction: *decoder* $d(.) : \mathbb{R}^k \to \mathbb{R}^p$.
- Common criterion: minimising the mean squared reconstruction loss

$$\min \mathbb{E}\left[\|X - d(e(X))\|^2\right]$$

- Examples:
  - Principal Component Analysis (PCA): linear encoder and decoder
  - Autoencoders (AE): neural network encoder and decoder

Dimension reduction

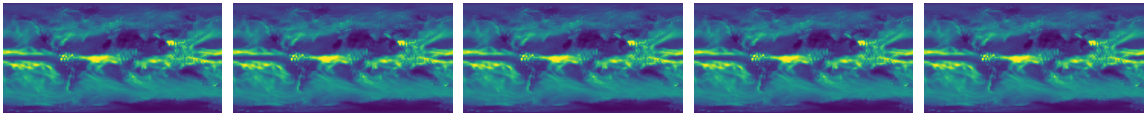
- Data $X \in \mathbb{R}^p$
- Dimension reduction: *encoder* $e(.) : \mathbb{R}^p \to \mathbb{R}^k$ where $k < p$.
- Data reconstruction: *decoder* $d(.) : \mathbb{R}^k \to \mathbb{R}^p$.
- Common criterion: minimising the mean squared reconstruction loss

$$\min \mathbb{E}\left[\|X - d(e(X))\|^2\right]$$

- Examples:
  - Principal Component Analysis (PCA): linear encoder and decoder
  - Autoencoders (AE): neural network encoder and decoder
- Lossy compression: when $k < p$, we typically have $X \neq d(e(X))$.

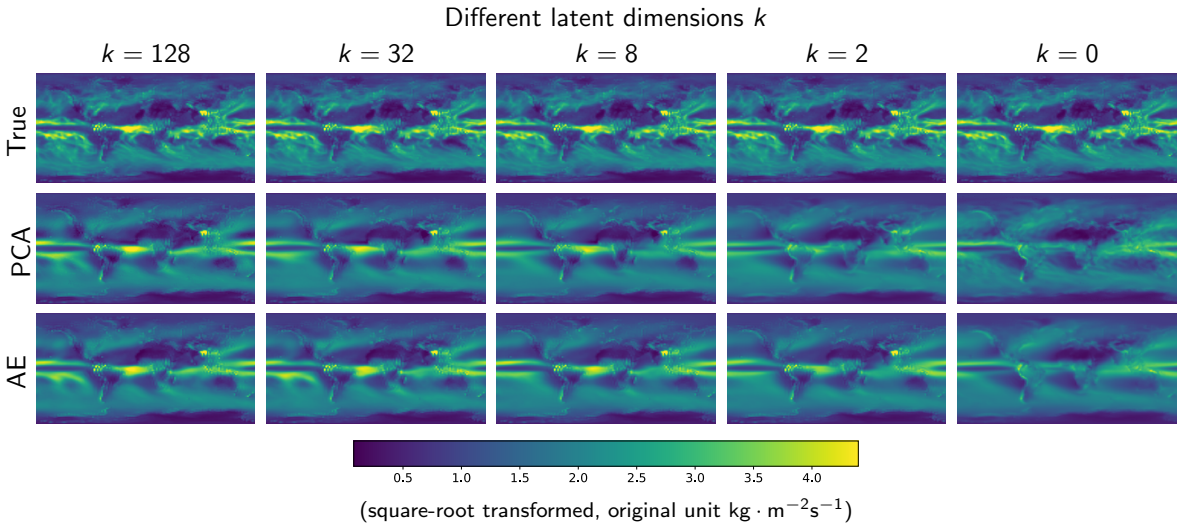
Reconstructions for global monthly precipitation fields with a spatial dimension of 360×180

True

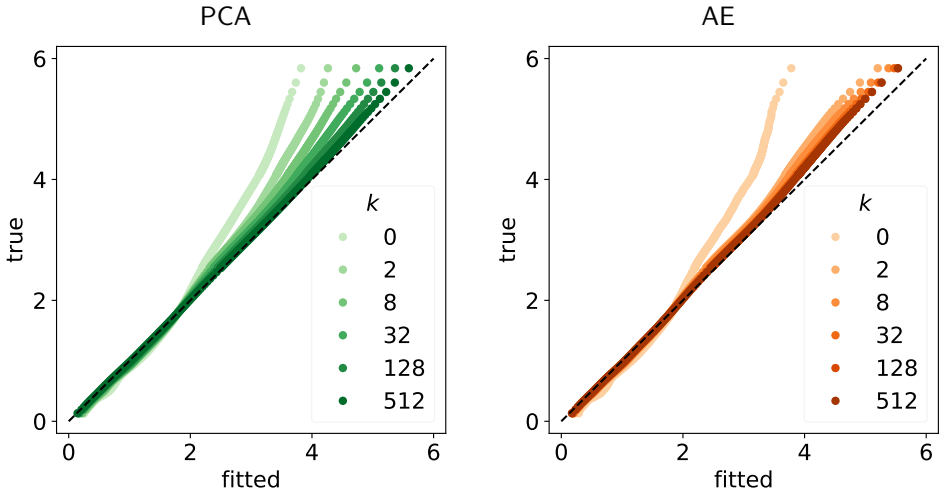


(square-root transformed, original unit $\mathrm{kg} \cdot \mathrm{m}^{-2}\mathrm{s}^{-1}$)

Reconstructions for global monthly precipitation fields with a spatial dimension of 360×180



Q–Q plots of precipitations at a random location for test data versus fitted distributions



Distributional reconstruction

- Mean reconstruction (autoencoders):

$$d(z) = \mathbb{E}[X|e(X) = z], \ \forall z.$$

Distributional reconstruction

- Mean reconstruction (autoencoders):

$$d(z) = \mathbb{E}[X|e(X) = z], \ \forall z.$$

- Distributional reconstruction (ours):

$$d(z, \varepsilon) \overset{d}{=} \big(X|e(X) = z\big), \ \forall z.$$

Distributional reconstruction

- Mean reconstruction (autoencoders):

$$d(z) = \mathbb{E}[X|e(X) = z], \ \forall z.$$

- Distributional reconstruction (ours):

$$d(z, \varepsilon) \overset{d}{=} \big(X|e(X) = z\big), \ \forall z.$$

$\Rightarrow$ Distributionally lossless compression:

$$d(e(X), \varepsilon) \overset{d}{=} X$$

irrespective of the latent dimension.

Distributional Principal Autoencoder (DPA)

To achieve distributional reconstruction, i.e.

$$d(z, \varepsilon) \overset{d}{=} \big(X | e(X) = z\big), \ \forall z.$$

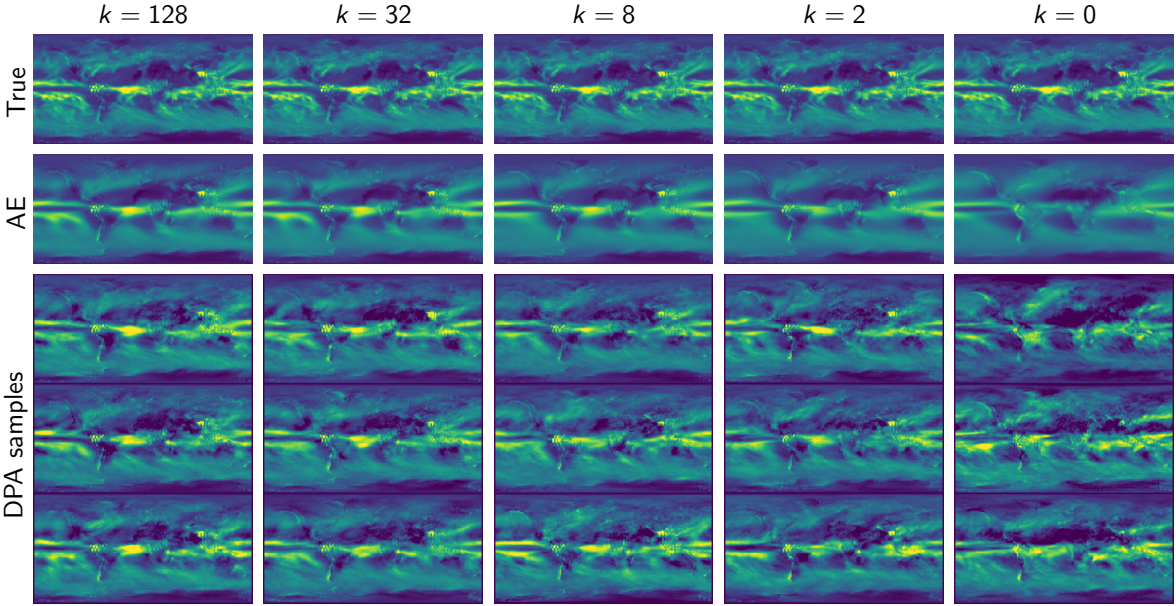
Distributional Principal Autoencoder (DPA)

To achieve distributional reconstruction, i.e.

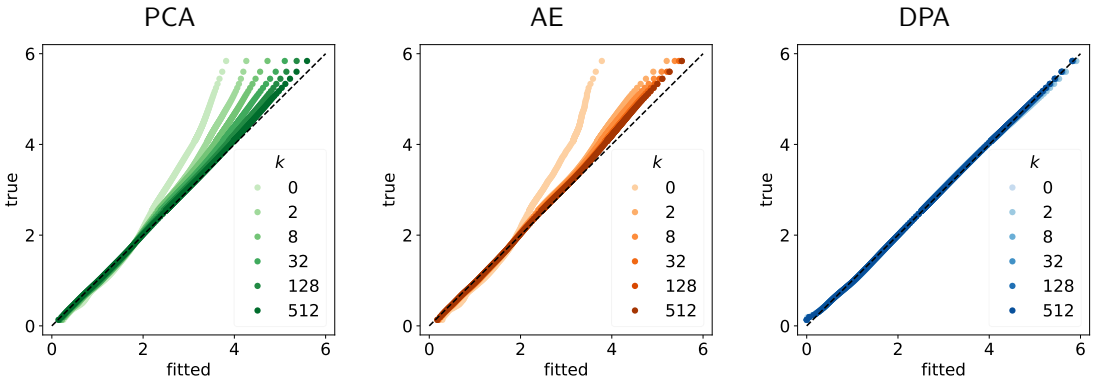
$$d(z, \varepsilon) \overset{d}{=} \big(X | e(X) = z\big), \ \forall z.$$

DPA solution (engression applied to $X|e(X)$):

$$\underset{e,d}{\operatorname{argmin}} \mathbb{E}\Big[\|X - d(e(X), \varepsilon)\| - \frac{1}{2}\|d(e(X), \varepsilon) - d(e(X), \varepsilon')\|\Big]$$


$k = 128$
$k = 32$
$k = 8$
$k = 2$
$k = 0$
True
AE
DPA samples

Q–Q plots of precipitations at a random location for test data versus fitted distributions



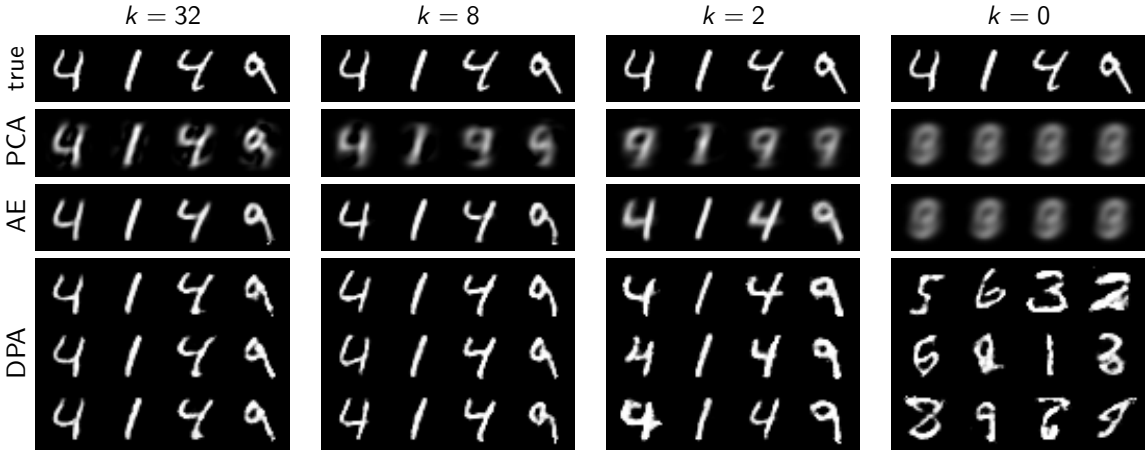


Figure: Reconstructions for MNIST. 28×28 hand-written digits.

Summary III

Unsupervised engression for $X|e(X) \Rightarrow$ distributionally lossless dimension reduction

Summary

- A distributional learning method *engression*
- Scientific applications:
  - Climate science: statistical emulation of physical climate models
  - Single-cell genomics, proteomics: prediction for unseen perturbations
- Statistical problems:
  - Extrapolation in nonparametric regression (engression + pre-ANM)
  - Distributionally lossless dimension reduction (unsupervised engression for $X|e(X)$)
  - Causal effect estimation (instrumental variable engression)
  - Robust prediction under distribution shifts (multi-environment, invariant engression)[1]

[1] Henzi, S., Law, Bühlmann, "Invariant Probabilistic Prediction," *Biometrika*, 2024+

Outlook

- For statisticians, engression provides a flexible tool for statistical inference problems that
  - involve distribution estimation or
  - require stronger identification condition.
- For applied researchers, engression can be an addition to the current data analysis toolkit.
  - comprehensive distributional information
  - different extrapolation behavior

Thank you!


X. Shen and N. Meinshausen, "Engression: Extrapolation from the Lens of Distributional Regression," *Journal of the Royal Statistical Society: Series B*, 2024+.


X. Shen and N. Meinshausen, "Distributional Principal Autoencoders," *arXiv preprint arXiv:2404.13649*, 2024.


A. Henzi, X. Shen, M. Law, and P. Bühlmann, "Invariant Probabilistic Prediction," *Biometrika*, 2024+.


A. Holovchak, S. Saengkyongam, N. Meinshausen, and X. Shen, "Distributional Instrumental Variable Regression," 2024+.