Generalization Beyond Observations: Distribution is All You Need?

Xinwei Shen

Department of Statistics, University of Washington

October 28, 2025

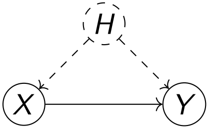
Classical setting

- Predictors $X \in \mathbb{R}^p$, response $Y \in \mathbb{R}^d$, training data $(X, Y) \sim P_{XY}$
- Target: functionals of conditional distribution $P^{\text{test}}_{Y|X=x}$, e.g., conditional mean/quantiles
- Classical setting: $P^{\text{test}}_X = P_X$ and $P^{\text{test}}_{Y|X=x} = P_{Y|X=x}$
- Method: fit the target on training data by empirical risk minimization (ERM)

Potential problems

Cases where **a naive ERM fit on training data may not be optimal:**

- Out-of-support covariate shifts (aka, extrapolation): beyond the training support of P_X
- Conditional shifts: shifts in $P_{Y|X}$
- Causal effects: interventional distribution of the outcome Y given treatment X in the presence of latent confounders



Need to generalize beyond the observed data distribution P_{XY}.

A **distributional** perspective for generalization

- Observed data from P_{XY}
- Inferential target not a functional of P_{XY}. E.g., conditional mean function under distribution shifts, treatment effects...

Estimating the full observed data distribution *(which allows to exploit more information from data) for better generalization beyond it.*

Showcases:

- Out-of-support covariate shifts[1]
- Conditional shifts[2]
- Causal effect estimation[3]

[1] S. and Meinshausen, "Engression: Extrapolation through the Lens of Distributional Regression," *JRSSB*, 2024
[2] Henze, S., Law, and Bühlmann, "Invariant Probabilistic Prediction," *Biometrika*, 2024
[3] Holovchak, Saengkyongam, Meinshausen, S., "Distributional Instrumental Variable Method," arXiv:2502.07641

Goal of this talk

Develop **distributional** approaches that can generalize better beyond the observations.

How to estimate a distribution?

Distribution estimation in classical statistics

Random variables X and Y (X can be empty set)

Target: $P_{Y|X=x}$

Methods: kernel density estimation, quantile regression, distributional regression (Koenker '05; Meinshausen '06; Dunson et al. '07; Hothorn et al. '14), etc.

Distribution estimation in classical statistics

- Restrictive parametric assumptions
- High computational cost with large sample sizes
- Not scalable to high dimensional responses
- Sampling is nontrivial! MCMC

Generative AI

Same goal: to learn a distribution by generating new samples from it.

Methods: diffusion models, generative adversarial networks, etc.

Images generated by DALL-E 3 (openai.com)

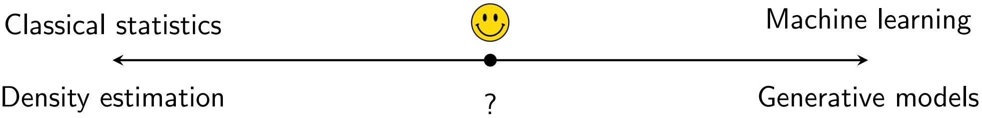
Excellent for images, texts, video.

What about scientific data, clinical data, etc?

Generative AI

- Computationally intensive — GPU time is all you need
- Hyperparameter tuning
- Emphasis on data generation rather than inference
- Not easy as a plug-in for our statistical procedures

Distribution learning



as simple as classical stat methods
as powerful as machine learning methods

Distributional learning via generative models

- Target: conditional distribution of $Y|X$
- Build a **generative model** to describe the distribution of $Y|X$:

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

  where $\varepsilon \sim P_\varepsilon$ pre-defined and map $g : (x, \varepsilon) \mapsto y$ is often parametrized by neural networks.
- Goal: find g such that $g(x, \varepsilon) \sim P_{Y|X=x}$ for any x
- Sampling-based inference: a model to sample from $P_{Y|X=x}$

Proper scoring rule

- Given a distribution P and an observation z, the **energy score**[1] is defined as

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

- Strictly proper scoring rule: 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 "$=$" $\Leftrightarrow P = P^\star$.

[1] Gneiting and Raftery, 2007

Our distributional learning method *engression*

- Population solution:

$$\tilde{g} \in \underset{g \in \mathcal{G}}{\operatorname{argmin}} \, \mathbb{E}_{(X,Y)\sim P}[-\mathrm{ES}(P_g(.|X), Y)]$$
$$= \underset{g \in \mathcal{G}}{\operatorname{argmin}} \, \mathbb{E}\Big[\|Y - g(X,\varepsilon)\|_2 - \frac{1}{2}\|g(X,\varepsilon) - g(X,\varepsilon')\|_2\Big]$$

  where $P_g(.|x)$ is the distribution of $g(x,\varepsilon)$ and $\varepsilon, \varepsilon'$ are independent draws from $\mathcal{N}(0, I)$.

- **Proposition**: under correct model specification, we have $\tilde{g}(x,\varepsilon) \sim P_{Y|X=x}$, $\forall x \in \mathrm{supp}(P_X)$.
- Algorithm: neural network $\mathcal{G}$, empirical risk, (stochastic) gradient descent.

Estimation of the functionals

Monte Carlo: for a fixed test point x,

1. Draw a sample of ε, i.e., $\varepsilon_1, \dots, \varepsilon_m$;
2. Then $\tilde{g}(x, \varepsilon_i)$, $i = 1, \dots, m$ is a sample of the estimated distribution of $Y|X = x$;
3. Obtain estimators:
   - 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[1]

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

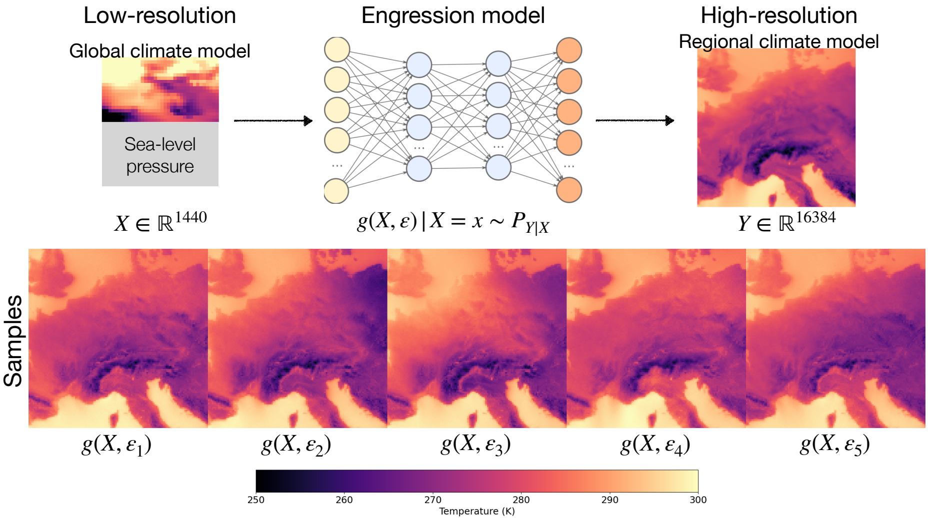
```
> library(engression)                                                       ## load engression package
> engressor = engression(X, Y)                                              ## fit an engression model
> predict(engressor, Xtest, type="mean")                                    ## mean prediction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

Python: `pip install engression`

```
> from engression import engression                                         ## load engression package
> engressor = engression(X, Y)                                              ## fit an engression model
> engressor.predict(Xtest, target="mean")                                   ## mean prediction
> engressor.predict(Xtest, target=[0.025, 0.5, 0.975])                      ## quantile prediction
> engressor.sample(Xtest, sample_size=100)                                  ## sampling
```

[1] http://github.com/xwshen51/engression

Engression for climate downscaling[1]



[1] Joint with M. Schillinger, M. Samarin, R. Knutti, and N. Meinshausen. arXiv:2509.26258

Classical statistics

Machine learning

Density estimation

Engression

Generative models

simple yet powerful tool
for distribution learning

Powerful compared to classical stat methods:

- capacity of neural networks alleviates limitations of parametric model specifications
- scalable to (very) high-dimensional X and Y
- no quantile crossing

Simple compared to modern generative models:

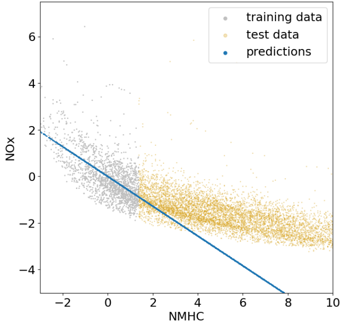
- computationally lighter:
  one-step sampling, no discriminator/minmax
- fewer tuning parameters
- focus on downstream estimation and inference

Problem I Out-of-support covariate shifts (extrapolation)[1]

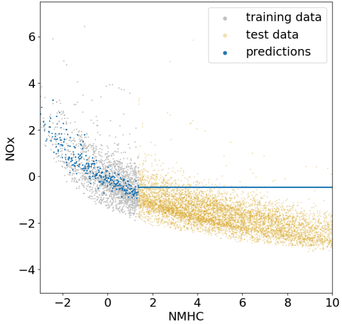
[1] S. and Meinshausen, "Engression: Extrapolation through the Lens of Distributional Regression," *JRSSB*, 2024

Challenge of nonlinear extrapolation

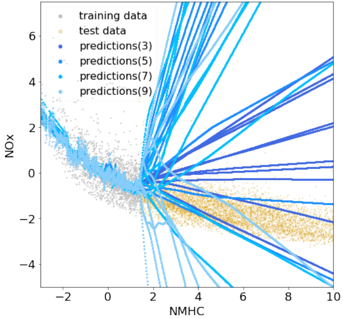
Air-quality data with measurements of two pollutants



Linear regression



Random Forests

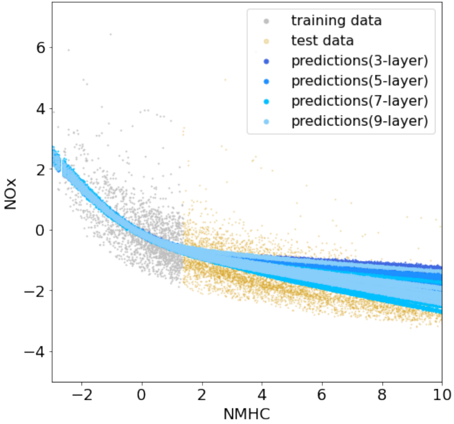


Neural network regression[1]

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

Engression makes a difference

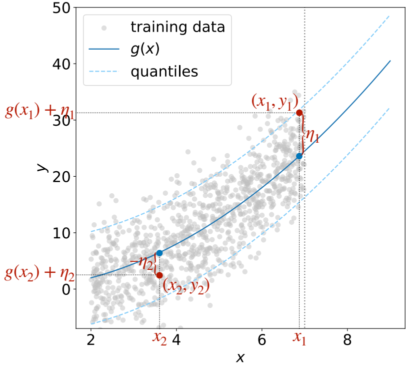
The reliability of engression does not break down immediately at the support boundary.



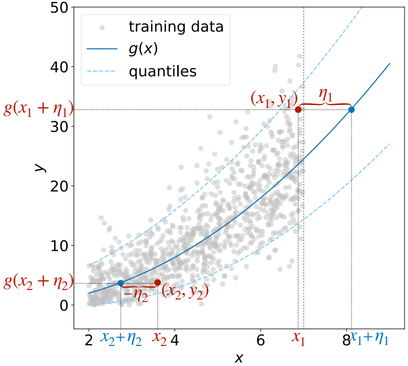
Results of engression with 3, 5, 7, or 9 layers and random initializations.

Additive noise models (ANMs)

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



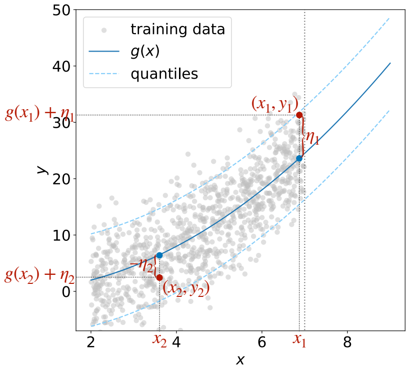
Pre-ANM: $Y = g(X + \eta)$



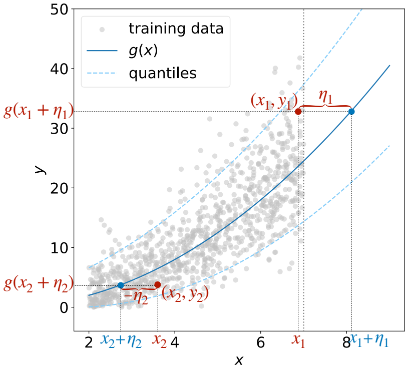
All models are wrong, but can one of them be useful in terms of extrapolation?

Additive noise models (ANMs)

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



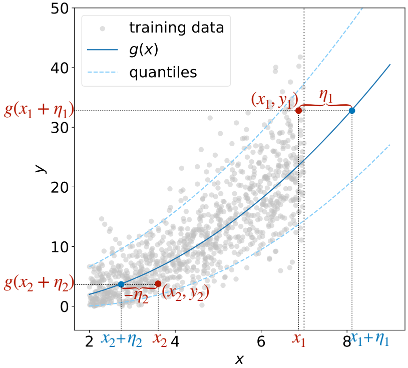
Pre-ANM: $Y = g(X + \eta)$



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

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.

Engression has the two ingredients for extrapolation

✓ Engression is **a distributional learning method**.

✓ Engression model class $\{g(x,\varepsilon)\}$ 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.

Regression fails to extrapolate

Theory setup:

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

Proposition (S. and Meinshausen, '24)

Let $\mathcal{F}_{L_1} := \operatorname{argmin}_{g \in \mathcal{G}} \mathbb{E}_{P_{\mathrm{tr}}} |Y - g(X)|$. 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

Theory setup:

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

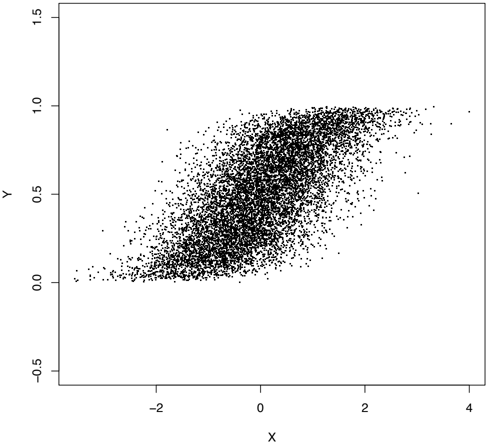
Theorem (S. and Meinshausen, '24)

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

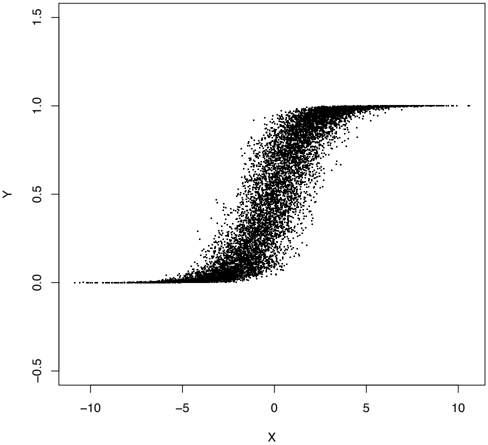
- Population engression $(\tilde{g}, \tilde{h})$ recovers the true model beyond the training support.
- Blessing of noise: the more (pre-additive) noise there is, the farther one can extrapolate.

Numerical example

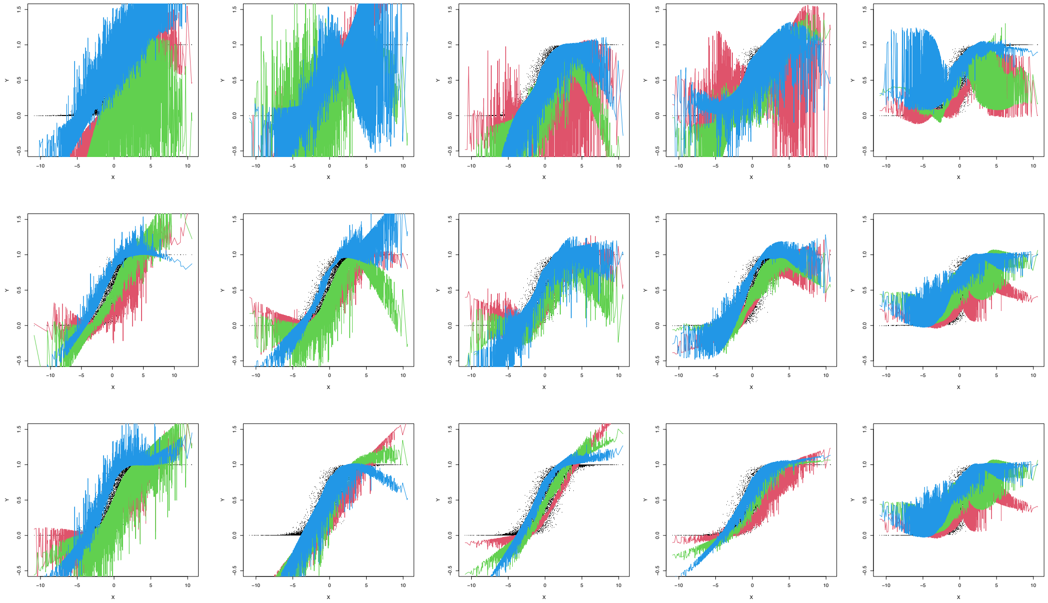
training data



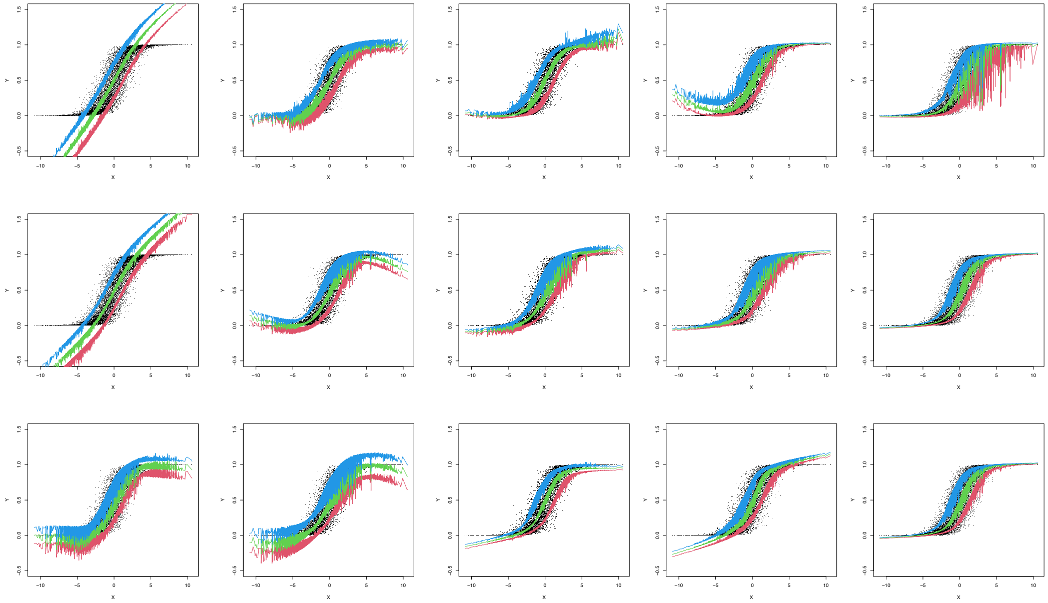
test data



NN quantile regression. Top to bottom: $10, 100$ and 1000 hidden dimension. Left to right: $2, 3, 5, 10$ and 20 layers.



Engression. Top to bottom: 10, 100 and 1000 hidden dimension. Left to right: 2, 3, 5, 10 and 20 layers.



Large-scale real-data experiments

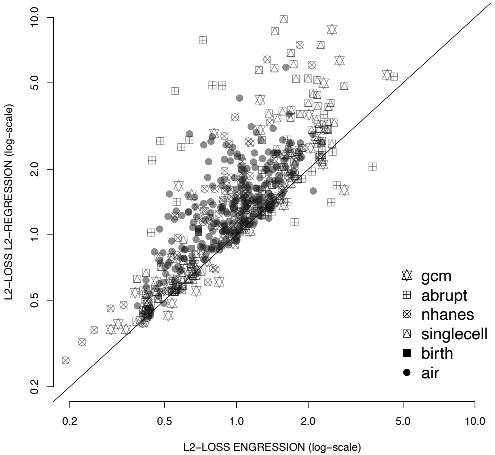
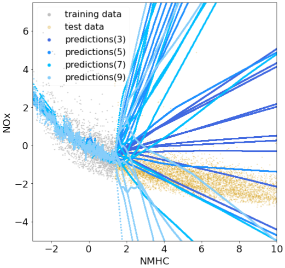


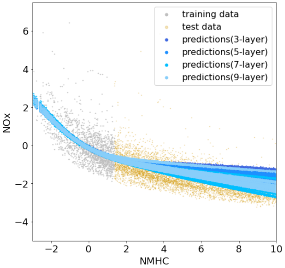
Figure: **Out-of-support losses** (in log-scale) of engression and regression for various data configurations, averaging over all hyperparameter settings.

Takeaway I

Engression + pre-ANM $\Rightarrow$ better extrapolation than standard regression



NN regression



engression

Problem II General shifts[1]

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

Invariant Probabilistic Prediction (IPP)

- Heterogeneous (multi-environment) data: for $e = 1, \ldots, m$,

$$\begin{aligned} X^e &= h^e(\varepsilon_X) \\ Y^e &= g^\star(X^e, \varepsilon_Y) \end{aligned}$$

- Given a proper scoring rule S and a model $P_\theta(y|x)$, "engression risk" per environment

$$\mathcal{R}_S^e(\theta) = \mathbb{E}[-S(P_\theta(y|X^e), Y^e)].$$

- Population **IPP** (invariant engression):

$$\min_\theta \frac{1}{m} \sum_{e=1}^{m} \mathcal{R}_S^e(\theta) + \lambda D(\mathcal{R}_S^1(\theta), \ldots, \mathcal{R}_S^m(\theta))$$

where $D(v) = \frac{1}{m^2} \sum_{i,j=1}^{m} (v_i - v_j)^2$, $\lambda \geq 0$ tuning parameter. $\lambda = 0$: naive engression.

Single-cell data application

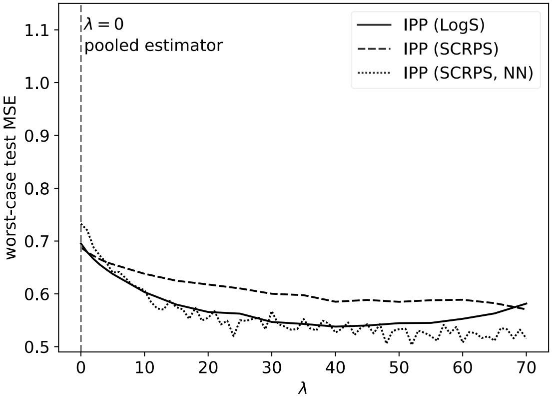
Single-cell RNA-sequencing data (Replogle et al. '22)

- 10 genes: a response and 9 predictors
- 10 training environments: 1 observational + 9 interventional
- 50 test environments that can be very different from training environments, due to interventions on unobserved genes

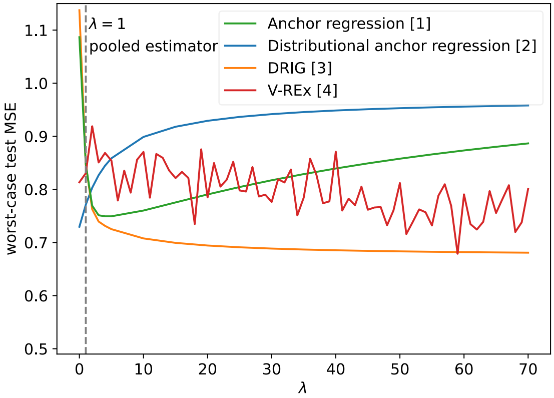
How robust our prediction model is on test environments?

Results on single-cell data

IPP



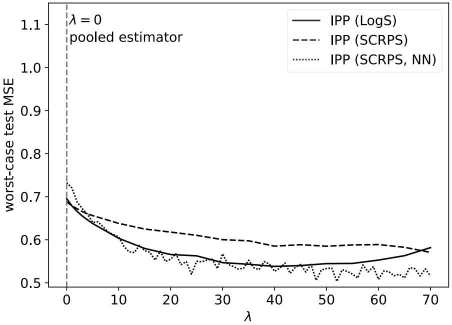
Competitors

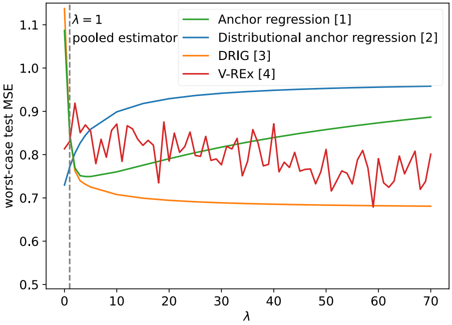


[1] Rothenhäusler et al. '21; [2] Kook et al. '22; [3] **S.**, Bühlmann, and Taeb, '23; [4] Krueger et al. '21

Takeaway II

Invariant distributional learning identifies a more robust prediction model than methods that only target robust mean prediction.





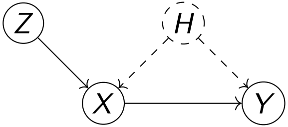
Problem III Causal effect estimation[1]

[1] Holovchak, Saengkyongam, Meinshausen, S., "Distributional Instrumental Variable Method," arXiv:2502.07641

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.

What would happen if everyone were given treatments $X = x$? i.e.

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

Identifiability

Theorem (HSMS** '25)**

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 $P_{\mathrm{obs}}(x, y|z)$.

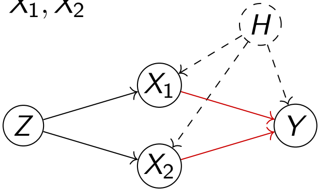
💡 Estimate $P_{\mathrm{obs}}(x, y|z) \overset{\text{sufficient}}{\Longrightarrow}$ identify $P(Y|do(X := x))$

necessary?

"Under-identified" case

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

$$X_1 = g_1(Z, \eta_1)$$
$$X_2 = g_2(Z, \eta_2)$$
$$Y = \beta_1 X_1 + \beta_2 X_2 + \eta_Y$$



- Two-stage least-squares (2SLS) 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 $P_{\text{obs}}(x_1, x_2, y|z)$.

Distributional instrumental variable (DIV) method

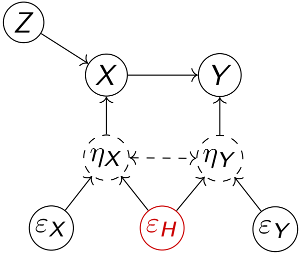
Joint generative model:

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

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

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

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



Distributional instrumental variable (DIV) method

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

$$\underset{f,g,h_X,h_Y}{\operatorname{argmin}}\ \mathbb{E}\left[\|(X,Y)-(\hat{X},\hat{Y})\|_2-\frac{1}{2}\|(\hat{X},\hat{Y})-(\hat{X}',\hat{Y}')\|_2\right],$$

where

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

Estimation of the interventional distribution and its functionals

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

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

- Estimation of the *interventional mean function*

$$\mu^*(x) := \mathbb{E}[f^*(x, h_Y^*(\varepsilon_Y, \varepsilon_H))].$$

  Average treatment effect: $\mu^*(x_1) - \mu^*(x_0)$

- Estimation of the *interventional quantile function*

$$q_\alpha^*(x) := Q_\alpha[f^*(x, h_Y^*(\varepsilon_Y, \varepsilon_H))].$$

  Quantile treatment effect: $q_\alpha^*(x_1) - q_\alpha^*(x_0)$

Simulation: interventional mean estimation in an 'under-identified' case

Setting: $Z \sim \mathrm{Unif}(-3, 3)$, $H, \varepsilon_X, \varepsilon_Y \sim \mathrm{Unif}(-1, 1)$ mutually independent, $\alpha \in \mathbb{R}$ is a tuning parameter; $X = Z(\alpha + 2H + \varepsilon_X)$, $Y = (1 + \exp(-(X + 2H + \varepsilon_Y)/3))^{-1}$.

It holds $\mathbb{E}(X|Z) = \alpha Z$ and $\mathrm{Var}(X|Z) = \frac{5}{3}Z^2$, where α controls the dependence of the conditional mean of $X|Z$.

	$\alpha = 0$	$\alpha = 1$	$\alpha = 5$
DIV	**0.002**	**0.002**	**0.002**
HSIC-X	2.693	0.333	0.344
CF linear	141.941	0.476	1.625
CF nonlinear	2.762	0.243	0.057
DeepGMM	1.158	0.274	0.005
DeepIV	0.675	0.305	0.102

Table: MSE of the estimated interventional mean functions.

Baselines: CF (Heckman, 76, Newey et al., 99, Guo & Small, 16); DeepIV (Hartford et al., 17); DeepGMM (Bennett et al., 20); HSIC-X (Saengkyongam et al., 22)

Takeaway III

Distributional causal learning can identify the causal effects in more cases than 2SLS.

Summary

Statistical **Estimating the full distribution** *for better generalization beyond observed distributions*

- Engression + pre-ANM $\Rightarrow$ out-of-support covariate shifts (better extrapolation than regression)[1]
- Multi-environment, invariant engression $\Rightarrow$ general shifts (more robust than methods that only target the mean)[2]
- Engression + instrumental variable $\Rightarrow$ causal effect estimation (more identification than 2SLS)[3]

Algorithmic How? Engression—simple yet powerful **generative AI** tool

Beyond generalization Have a problem involving distribution estimation? Try engression! *Thank you!*

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

[2] Henze, S., Law, and Bühlmann, "Invariant Probabilistic Prediction," *Biometrika*, 2024

[3] Holovchak, Saengkyongam, Meinshausen, and S., "Distributional Instrumental Variable Method," arXiv:2502.07641