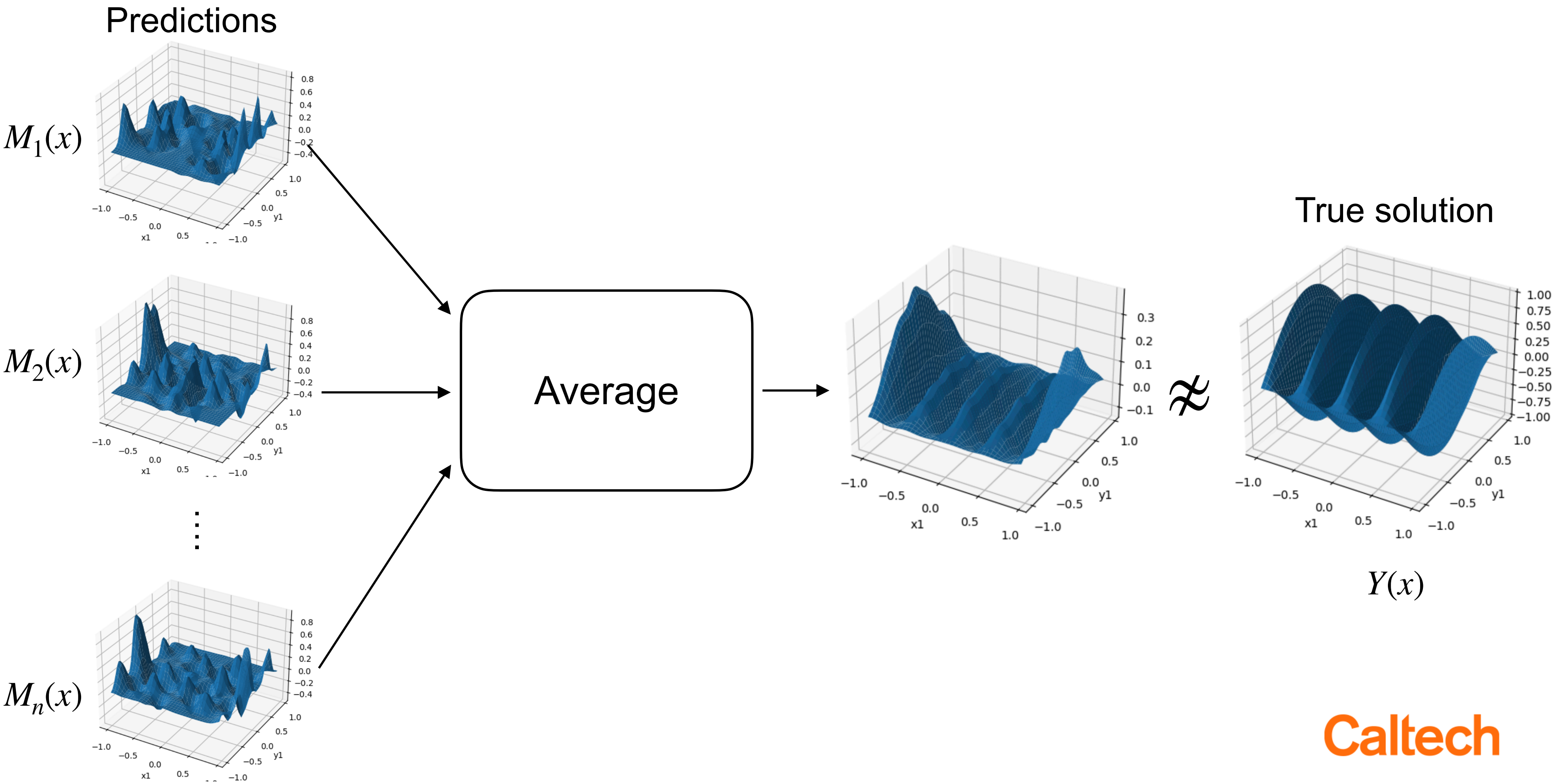
Model Aggregation: Data-driven combination of any prediction

Theo Bourdais

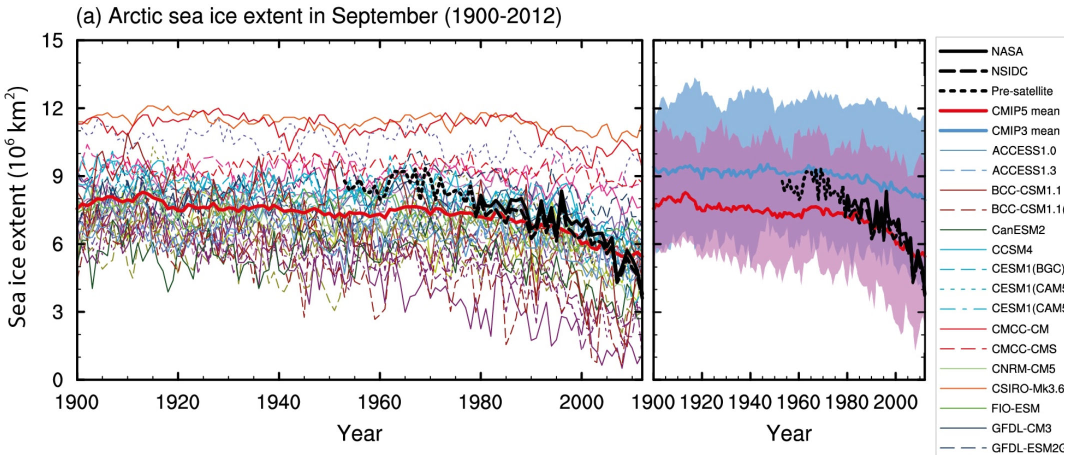
PhD Student, Computing + Mathematical Sciences

November 8th 2024

Caltech


Predictions
$M_1(x)$
$M_2(x)$
$M_n(x)$
Average
$\not\approx$
True solution
$Y(x)$

Real-life example from the IPCC



Arctic sea ice extent estimated by many models,

Coupled Model Intercomparison Project (report AR5 - figure 9.24)

Caltech

Existing aggregation methods

Trainable models

- **Average** (Random forests, bagging)
- **Trainable fixed combination** (Gradient boosting)
- **Trainable input dependant combination** (Mixture-of-experts)

Fixed models

- **Average**
- **Pick the best model**

There is a gap in refinement

The aggregation problem

Given the models,

$M_1(x)$

$\vdots$

$M_n(x)$

create a combination

(i.e. aggreation)

$$f(x, M_1(x), \ldots, M_n(x)) \approx Y(x)$$

that approximates the target

$\{X_i, Y(X_i)\}_{i=1}^N$ using data

Caltech

Best Mean Squared Error Aggregation

The best possible aggregation in Mean Squared Error is

$$M_A^*(x) := \underset{f \text{ measurable}}{\operatorname{argmin}} \mathbb{E}[|Y(x) - f(x, M_1(x), .., M_n(x))|^2] = \mathbb{E}[Y(x)|M_1(x), ..., M_n(x)]$$

This is intractable in general

Special Case: $(Y(x), M_1(x), .., M_n(x))$ **is Gaussian**

$$M_A^*(x) = \sum_{i=1}^{n} \alpha_i^*(x) M_i(x)$$

$$\alpha^*(x) = \underset{a \in \mathbb{R}^n}{\operatorname{argmin}} \mathbb{E}\left[\left|Y(x) - \sum_{i=1}^{n} a_i M_i(x)\right|^2\right] = \mathbb{E}\left[M(x)M(x)^T\right]^{-1} \mathbb{E}\left[M(x)Y(x)\right]$$

Caltech

Best case aggregation: Gaussian models

To solve the Laplace equation:

$$\begin{cases} \Delta Y = f & on\ \Omega \\ Y = g & on\ \partial\Omega \end{cases}$$

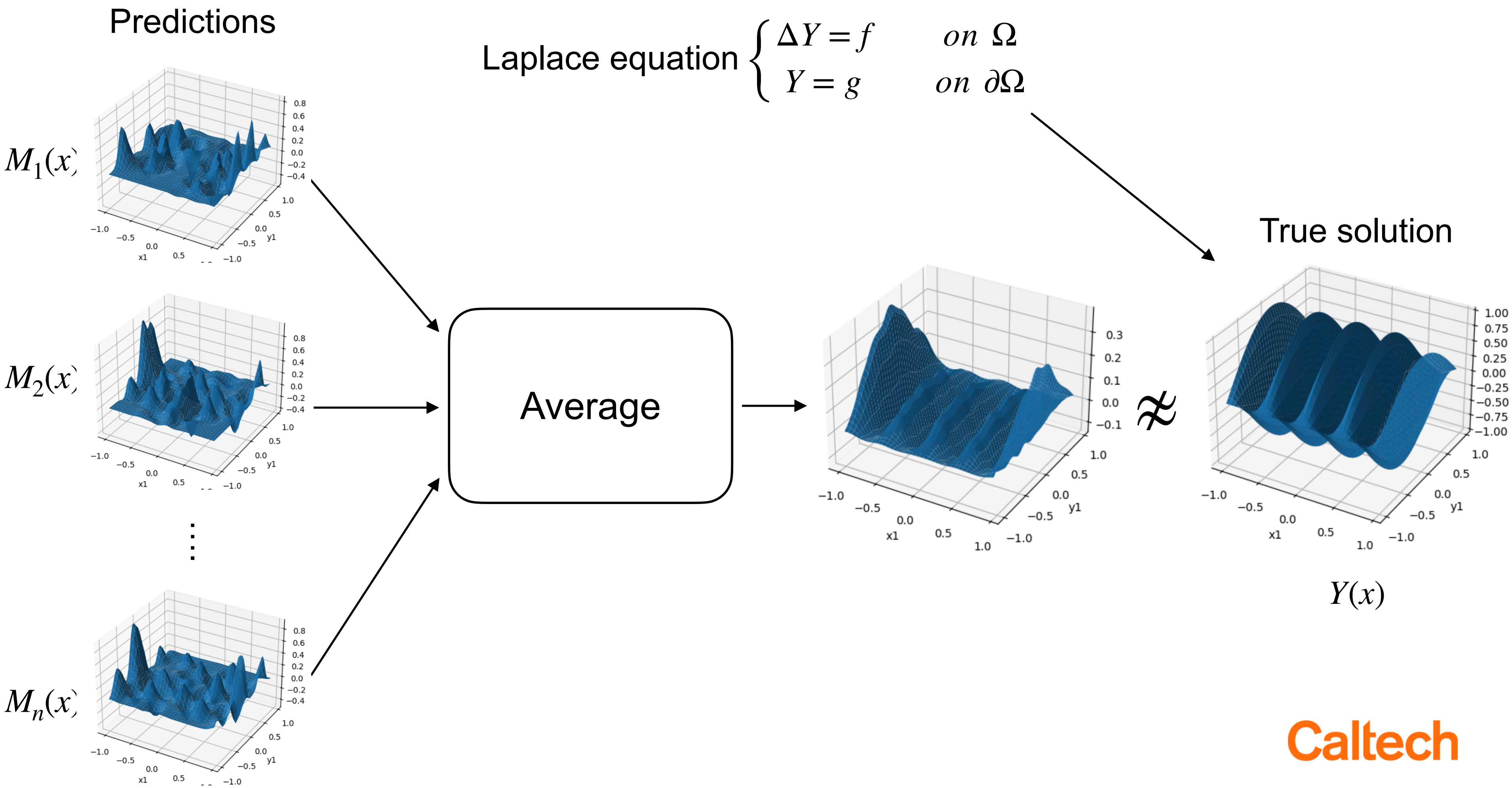
We can use a Gaussian process with:

- A kernel k
- A set of collocation points $X \subset \Omega$

To get a Gaussian approximation of the solution [Chen et al., 2021]

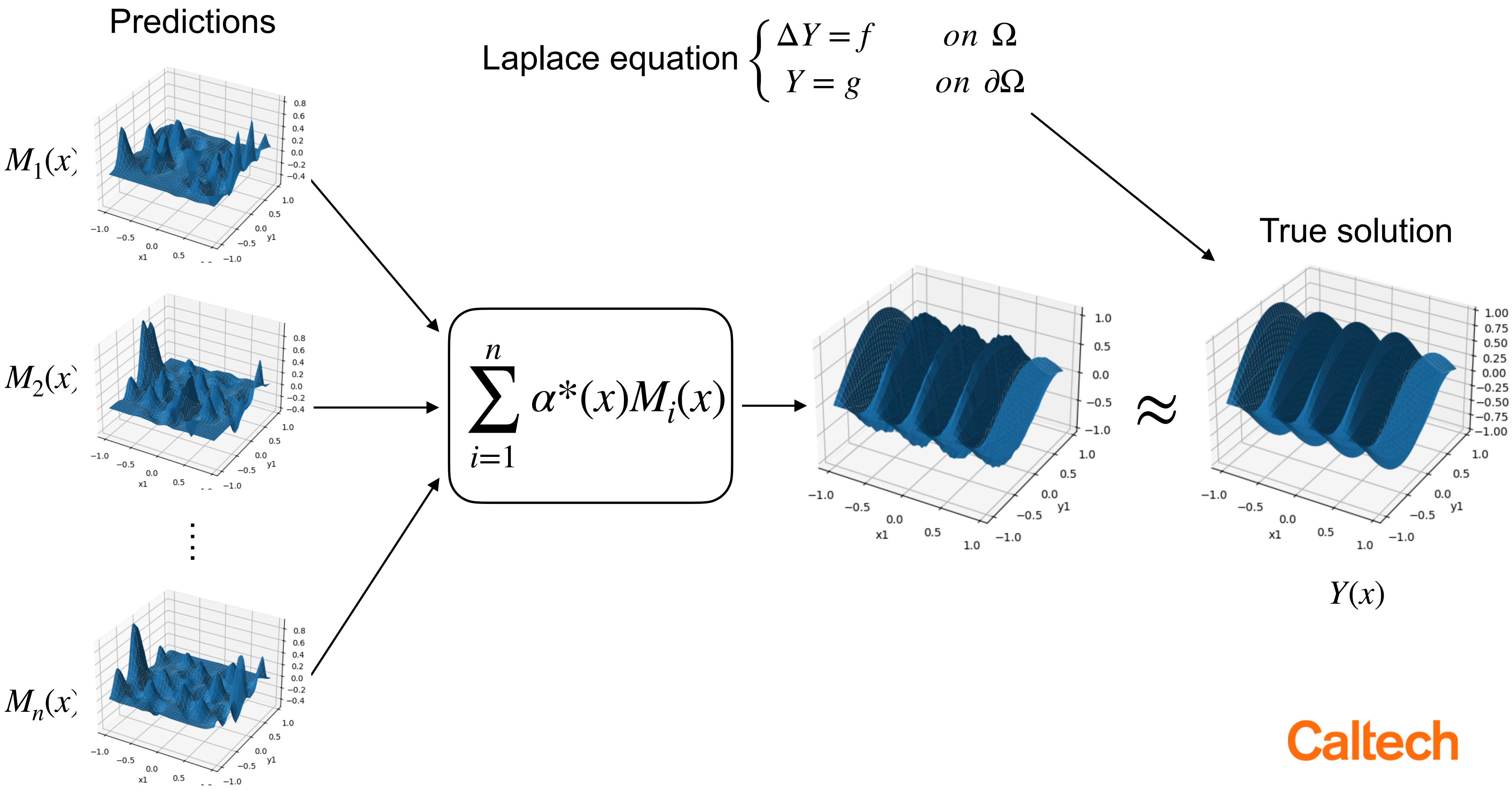
$$\xi \sim \mathcal{N}(0,k), \hat{Y} = \mathbb{E}[\xi \mid \Delta\xi(X) = f(X)]$$

Caltech



Predictions
Laplace equation $\begin{cases} \Delta Y = f & on\ \Omega \\ Y = g & on\ \partial\Omega \end{cases}$
$M_1(x)$
$M_2(x)$
$M_n(x)$
Average
$\not\approx$
True solution
$Y(x)$
x1
y1

Caltech


Predictions
$M_1(x)$
$M_2(x)$
$M_n(x)$
Laplace equation $\begin{cases} \Delta Y = f & on\ \Omega \\ Y = g & on\ \partial\Omega \end{cases}$
$\sum_{i=1}^{n} \alpha^*(x) M_i(x)$
$\approx$
True solution
$Y(x)$
x1
y1
Caltech

Minimal Error Aggregation

α^* is defined as:

$$\alpha^*(x) = \underset{a \in \mathbb{R}^n}{\operatorname{argmin}} \, \mathbb{E}\left[\left| Y(x) - \sum_{i=1}^{n} a_i M_i(x) \right|^2\right]$$

And we only have access to data $\{X_i, Y(X_i)\}_{i=1}^{N}$. So we need to learn over the training set and extrapolate for all x

$$\hat{\alpha}_E = \underset{a}{\operatorname{argmin}} \sum_{k=1}^{N}\left[\left| Y(X_k) - \sum_{i=1}^{n} a_i(X_k) M_i(X_k) \right|^2\right]$$

This does not work!

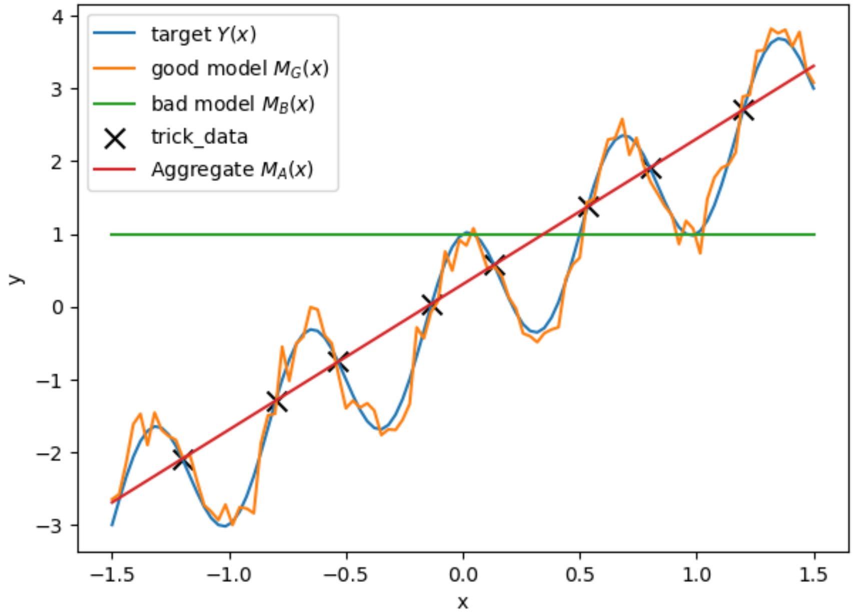
α is your favorite Machine Learning method (neural network, linear regression..)

Caltech

A pathological example

Given the target, models and data:

- Take α linear
  $\alpha(x) = (a_G x + b_G, a_B x + b_B)$
- Train using empirical MSE



Notice that:

- For each data point, the good model performs better than the bad model
- The best aggregation ignores the bad model

- The aggregate ignores the good model and interpolates the data
- Aggregation uses models as features, not approximations of Y

Caltech

Minimal Variance Aggregation

Let:

$$\begin{cases} M_1(x) = Y(x) + \epsilon_1(x) \\ \vdots \\ M_n(x) = Y(x) + \epsilon_n(x) \end{cases}$$

Where

- ϵ_i are independent (ease of presentation)
- We write $\mathbb{E}\left[|Y(x) - M_i(x)|^2\right] = V_i(x)$

Caltech

Minimal Variance Aggregation

We need to define what a good model is

$$\begin{cases} M_1(x) = Y(x) + \epsilon_1(x) \\ \vdots \\ M_n(x) = Y(x) + \epsilon_n(x) \end{cases} \quad \text{where} \quad \begin{cases} \text{(For simplicity)} & \epsilon_i \text{ are independent} \\ \text{(Write)} & Var[\epsilon_i(x)] = V_i(x) \\ \boxed{\text{(Assumption)} \quad \mathbb{E}[\epsilon_i(x)] = 0} \end{cases}$$

Then the Best Linear Unbiased Estimator (BLUE) of Y is:

$$\alpha_V(x)^T M(x) = \frac{\sum_{i=1}^{n} \frac{1}{V_i(x)} M_i(x)}{\sum_{i=1}^{n} \frac{1}{V_i(x)}}$$

No-bias assumptions helps define what a good/bad model is

Minimal Variance Aggregation

Problem: we don't have enough constraints

Add

$$\sum_{i=1}^{n} \alpha_i = 1$$

$$\alpha_V(x) = \underset{\sum_{i=1}^{n} a_i = 1}{\operatorname{argmin}} \; \mathbb{E}\left[\left| Y(x) - \sum_{i=1}^{n} a_i M_i(x) \right|^2\right] \quad \Longrightarrow \quad \alpha_V(x)^T M(x) = \frac{\sum_{i=1}^{n} \frac{1}{V_i(x)} M_i(x)}{\sum_{i=1}^{n} \frac{1}{V_i(x)}}$$

We just need to learn $V_i(x)$

Learning the variance/error

To predict the variance, we:

- Write $V_i(x) = e^{\lambda_i(x)}$ where λ_i is a Machine Learning method (Gaussian process, neural network…) to ensure positivity
- Use the loss

$$\min_{\lambda_i \in \mathscr{H}} \sum_{k=1}^{N} \Big[\underbrace{e^{\lambda_i(X_k)}}_{V_i(X_k)} - \underbrace{(Y(X_k) - M_i(X_k))^2}_{\text{Empirical variance}}\Big]^2 + \underbrace{\eta \|\lambda_i\|_{\mathscr{H}}^2}_{\text{Regularization}}$$

Caltech

Theorem on linear regression:

Assume samples $(M_j, Y_j)_{j=1}^N$, which one has the best loss $\mathscr{L}(\alpha) = \mathbb{E}[|Y - \alpha^T M|^2]$?

$$\hat{\alpha}_E(x) = \operatorname*{argmin}_{a \in \mathbb{R}^n} \sum_{j=1}^{N} \left[\left| Y_j - a^T M_j \right|^2 \right]$$

Minimal (Empirical) Error Aggregation

$$\mathscr{L}(\hat{\alpha}_E) = \mathscr{L}(\alpha^*) + \mathcal{O}\left(\frac{1}{\sqrt{N}}\right)$$

$$\hat{\alpha}_V(x) = \operatorname*{argmin}_{a \in \mathbb{R}^n} \begin{cases} \sum_{j=1}^N \left[\left| Y_j - a^T M_j \right|^2 \right] \\ \text{such that } \sum_{i=1}^n a_i = 1 \end{cases}$$

Minimal (Empirical) Variance Aggregation

There exists $\lambda \in [0,1]$ s.t.:

$$\mathscr{L}(\hat{\alpha}_V) = \frac{1}{\lambda} \mathscr{L}(\alpha^*) + \mathcal{O}\left(\frac{1}{N}\right)$$

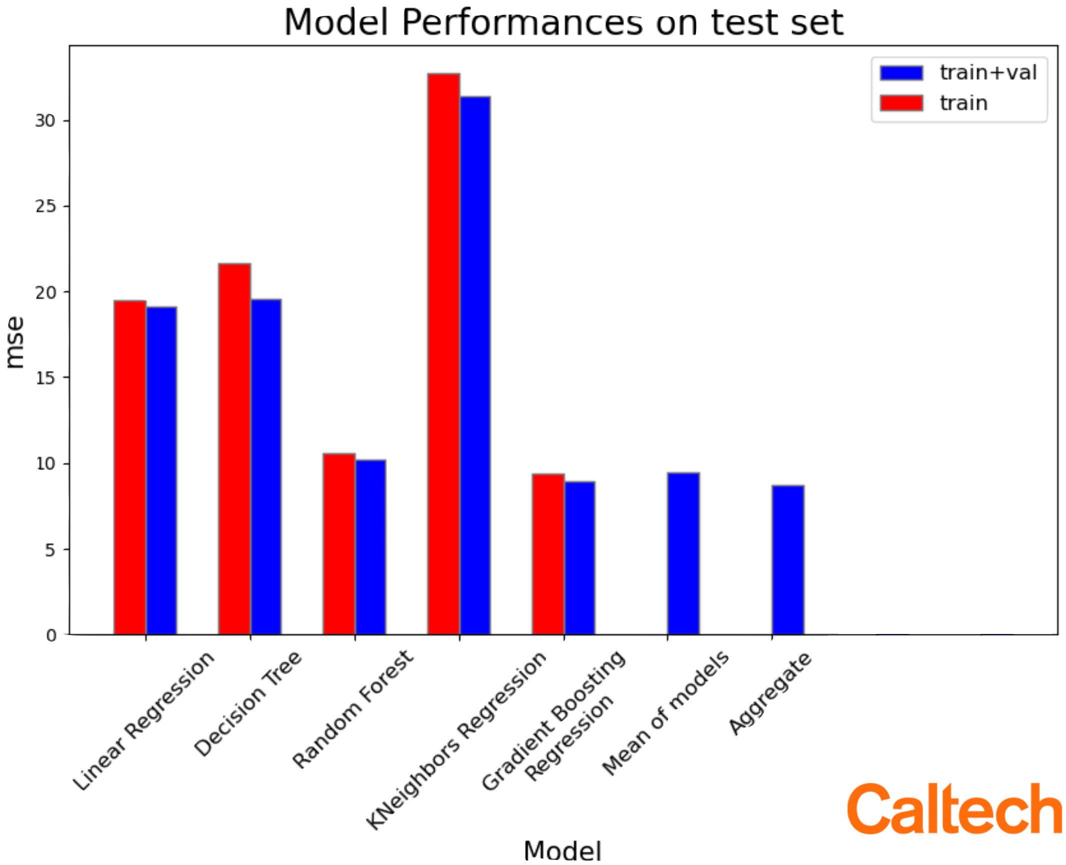
In model aggregation, N is small and $\lambda \to 1$

Applications

The Boston housing dataset

- Data: 506 samples $\{X_i, Y_i\}$
- Data is split into train-test-val
- Aggregation of red models using val data
    - Red models only see train data
    - Blue models for comparison see train+val

- Aggregation is:
    - Better than models aggregated
    - Better than the mean
    - Better than all models

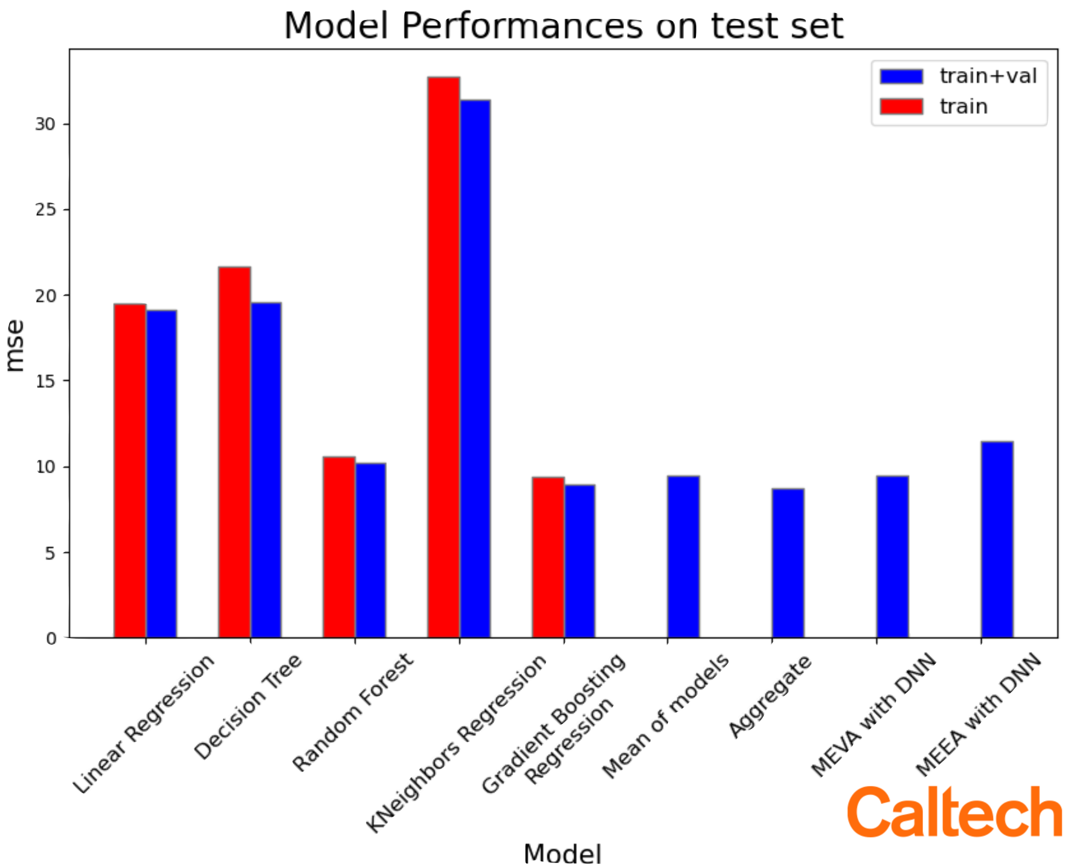


Caltech

The Boston housing dataset

A comparison with minimal error aggregation:

- Take two identical Neural networks
- Train:
    - To minimize error (bad loss)
    - To estimate variance (our loss)



Caltech

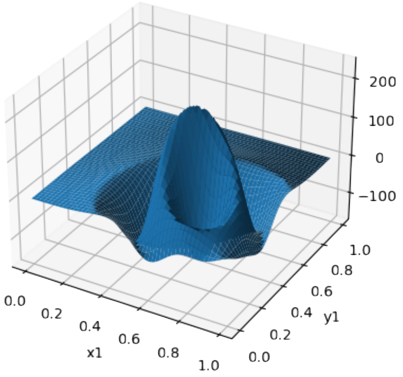
PDE examples

Given a PDE, we may have multiple solvers/approximations giving a solution.

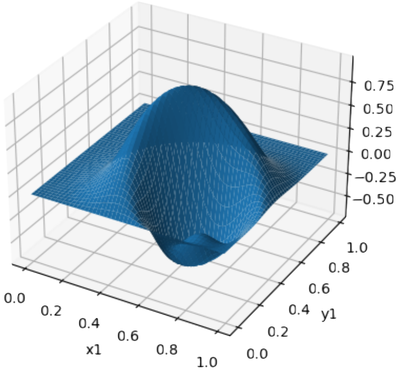
For example:

Laplace equation: $$\begin{cases} \Delta u = f & on\ \Omega \\ u = 0 & on\ \partial\Omega \end{cases}$$

Given models $M_i(f) \approx u$, we want to learn the aggregation operator $\alpha(f)$

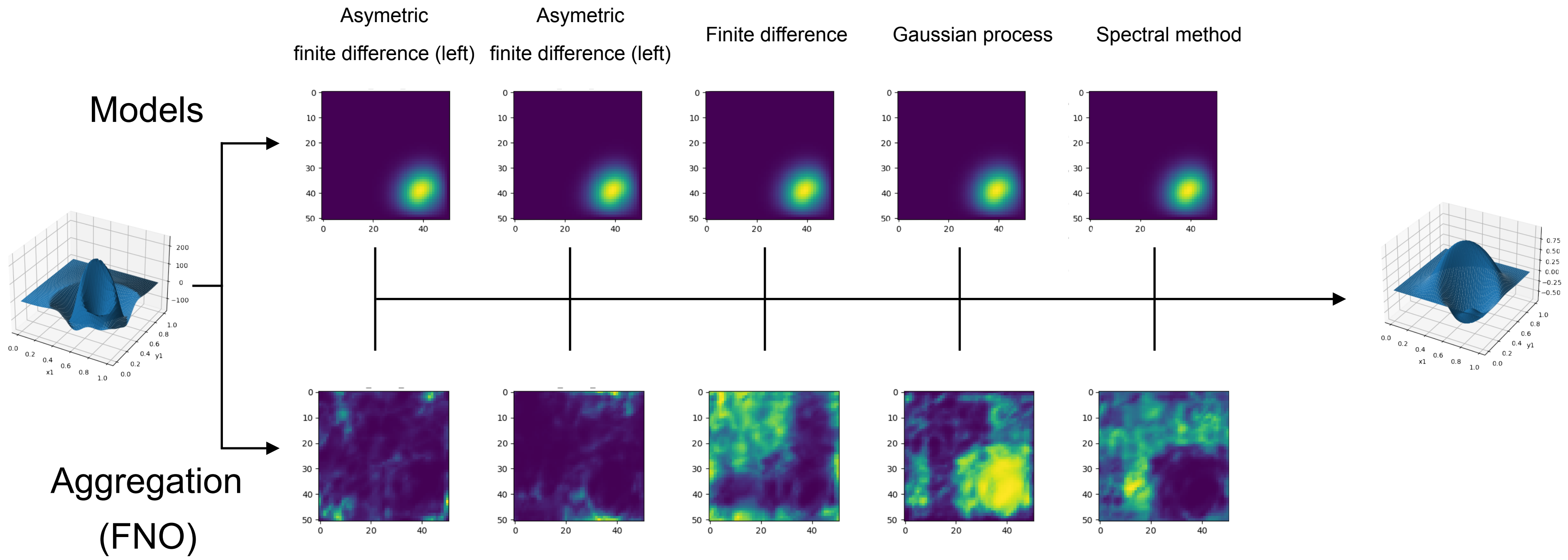


Random f



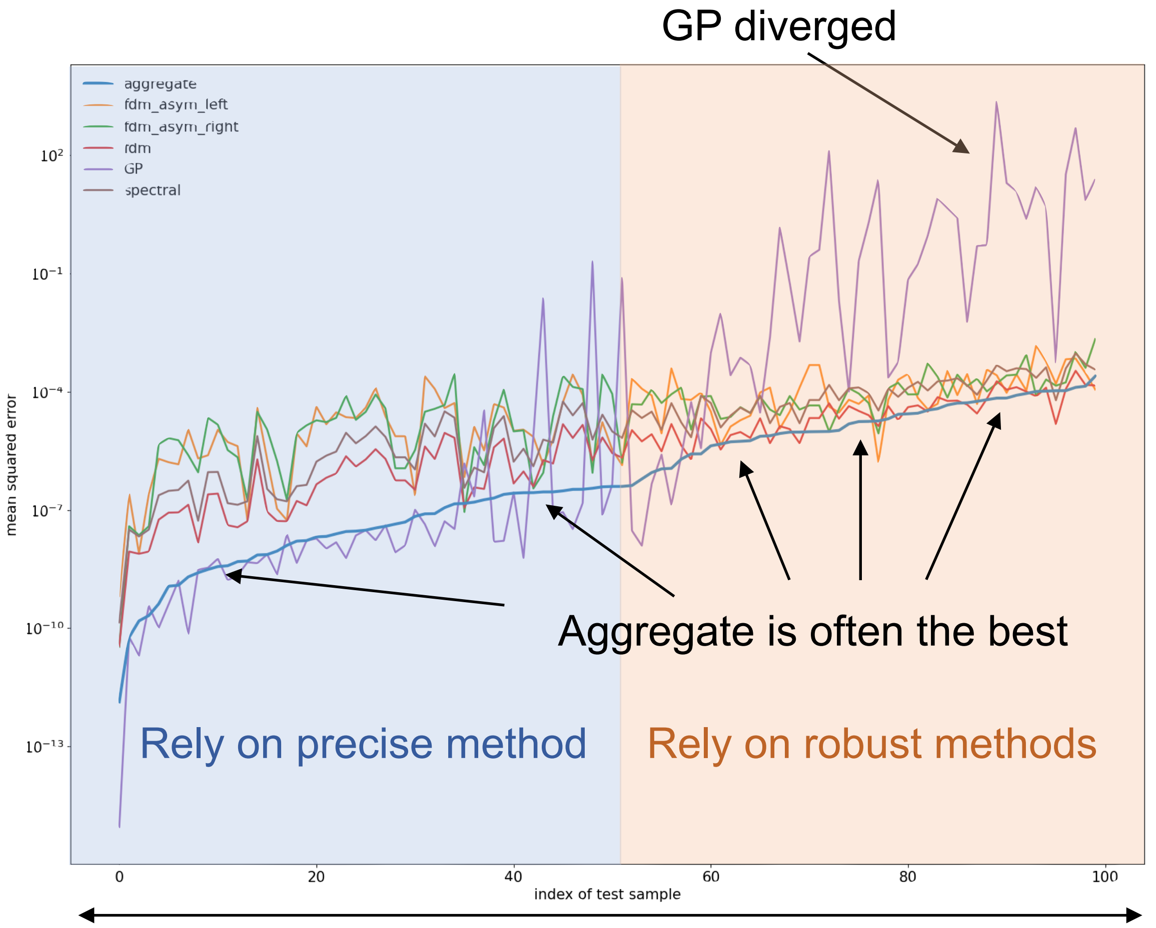
Random u

PDE example 1 - Laplace equation



Note: FNO uses both f and model outputs to predict aggregation

Caltech



Easy input Hard input

Method	Geometric mean of MSE (log scale)
Aggregate	**-6.282**
FDM	-5.523
Spectral	-4.988
Gaussian process	-4.739
FDM asymetric (right)	-4.685
FDM asymetric (left)	-4.699

PDE example 2 - Burger's equation

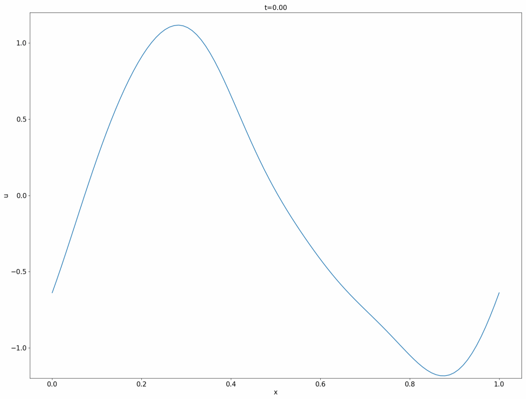
Consider Burger's equation on $\Omega = [0,1]^2$:

$$\begin{cases} \partial_t u + u\partial_x u = \nu \partial_{xx} u & \text{for} \quad (x,t) \in \Omega \\ u(0,x) = f(x) & \text{for} \quad x \in [0,1] \\ u(t,0) = u(t,1) & \text{for} \quad t \in [0,1] \end{cases}$$

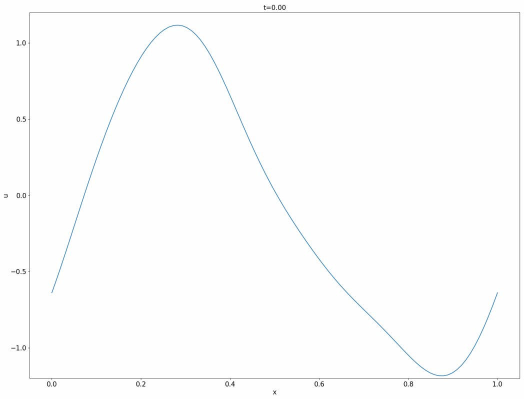
Choose:

- ν to be small
- $f \sim \mathcal{N}(0,K)$ where $K(x,y) = \exp\left(-\frac{2}{l^2}\sin^2\left(\pi|x_i - x_j|^2\right)\right)$
    - i.e. f is periodic and infinitely differentiable

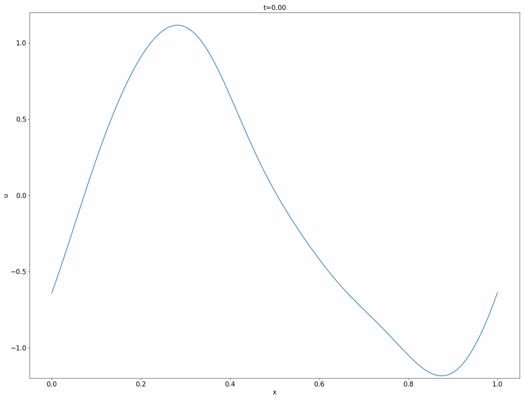
PDE example 2 - Burger's equation



Correct



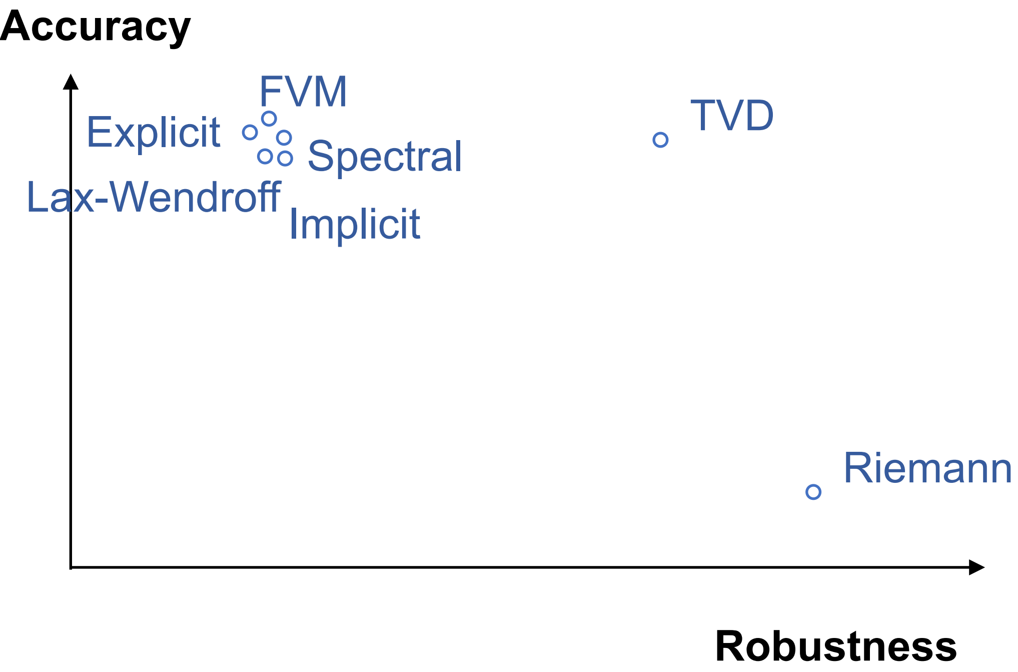
Blowup



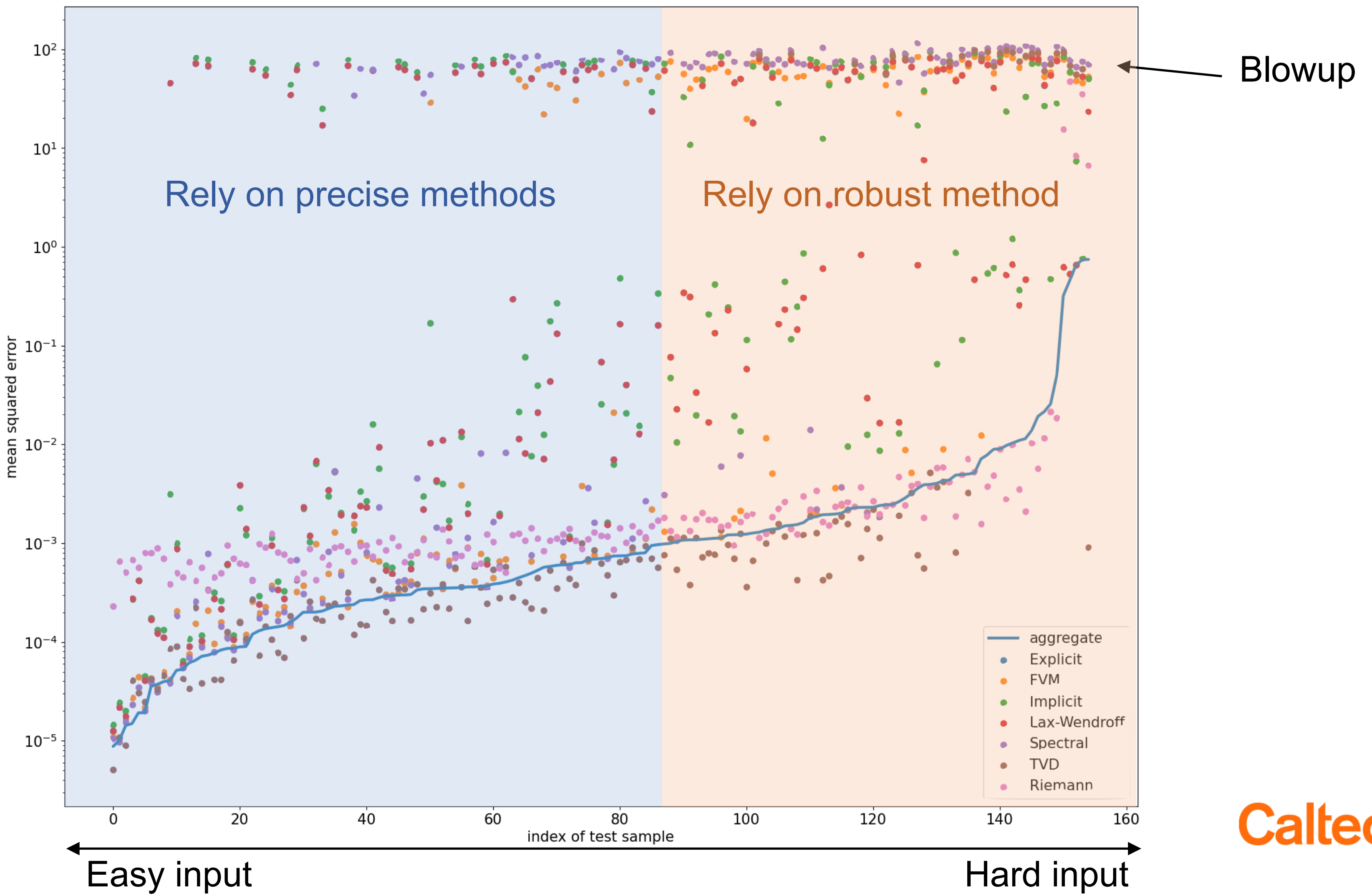
Oscillations

Caltech

PDE example 2 - Burger's equation



Method	Geometric mean of MSE (log scale)
Aggregate	**-3.106**
Riemann	-2.734
TVD	-2.568
FDM	-1.228
Spectral	-0.625
Implicit	-0.488
Explicit	-0.455
Lax-Wendroff	-0.455


Blowup
Rely on precise methods
Rely on robust method
mean squared error
index of test sample
aggregate
Explicit
FVM
Implicit
Lax-Wendroff
Spectral
TVD
Riemann
Easy input
Hard input

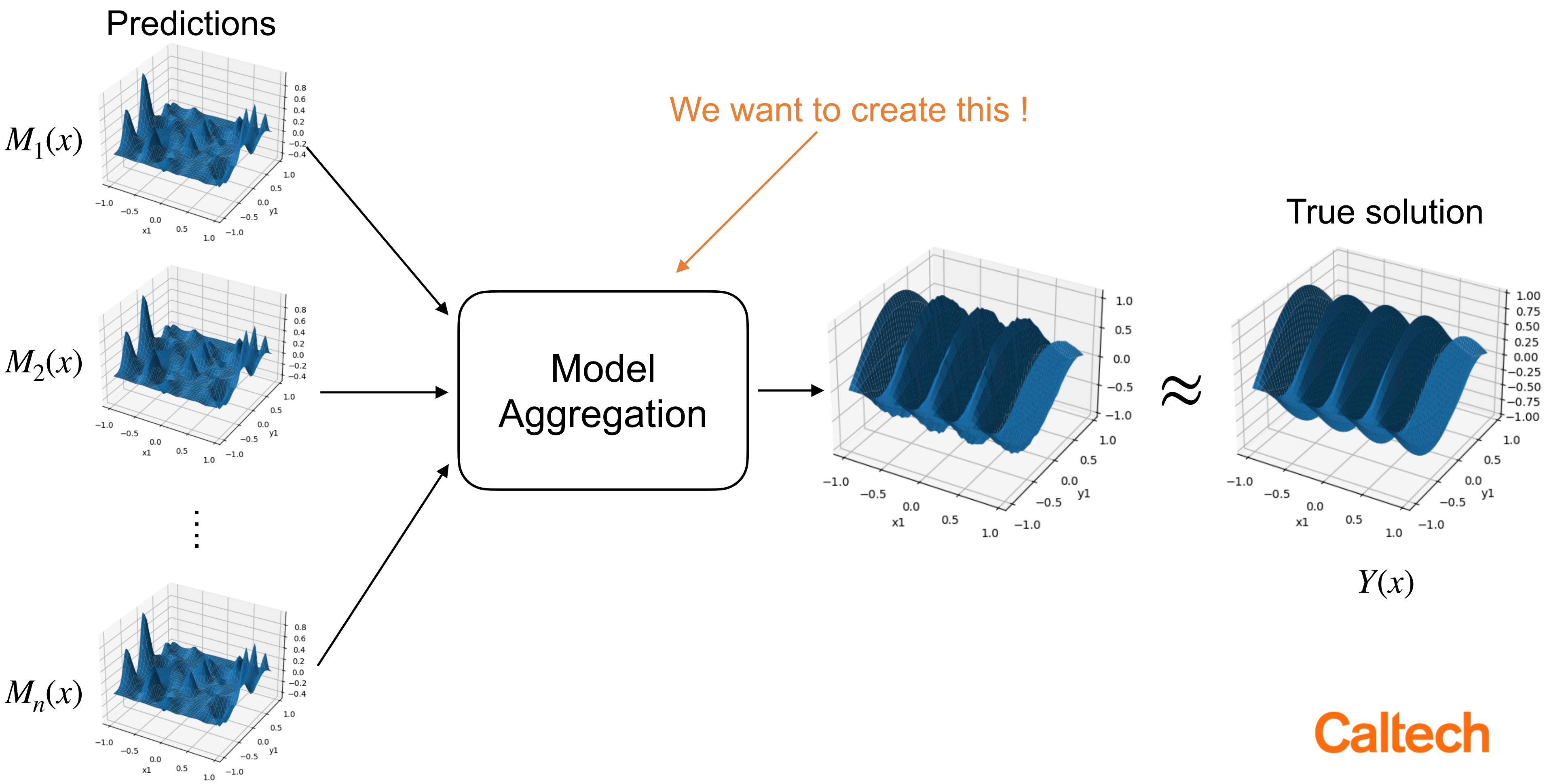
Caltech

Conclusion

We introduce a simple framework to aggregate existing models

- Only requires model output (no assumption, non intrusive)
- Most useful in scientific computing settings with legacy models
- Aggregate any type of methods (ML, solvers…)

Bourdais, T., & Owhadi, H. (2024).
Model aggregation: minimizing empirical variance outperforms minimizing empirical error
arXiv, accepted at ICLR2025



Predictions
$M_1(x)$
$M_2(x)$
$M_n(x)$
We want to create this !
Model Aggregation
$\approx$
True solution
$Y(x)$

Caltech

Summary

$$\text{Given} \quad \begin{matrix} M_1(x) \\ \vdots \\ M_n(x) \end{matrix} \longrightarrow f(x, M_1(x), \ldots, M_n(x)) \approx Y(x)$$

Caltech

Summary

1. Simplification + Gaussian ideal case

Given $M_1(x)$ $\vdots$ $M_n(x)$ $\rightarrow$ $\sum_{i=1}^{n} \alpha_i(x) M_i(x) \approx Y(x)$

Where

$$\alpha^*(x) = \operatorname*{argmin}_{a \in \mathbb{R}^n} \mathbb{E}\left[\left| Y(x) - \sum_{i=1}^{n} a_i M_i(x) \right|^2\right]$$

Caltech

Summary

Given

$$M_1(x) \quad \vdots \quad M_n(x)$$

No assumption

$$\sum_{i=1}^{n} \alpha_i(x) M_i(x) \approx Y(x)$$

Where

$$\alpha^* = \operatorname*{argmin}_{a} \sum_{k=1}^{N} \left[\left| Y(X_k) - \sum_{i=1}^{n} \alpha_i(X_k) M_i(X_k) \right|^2 \right]$$

$$\alpha^*(x) = \operatorname*{argmin}_{a \in \mathbb{R}^n} \mathbb{E}\left[\left\| Y(x) - \sum_{i=1}^{n} a_i M_i(x) \right\|^2 \right]$$

$$\{X_i, Y(X_i)\}_{i=1}^{N}$$

1. Simplification + Gaussian ideal case
2. Directly minimize error

Does not work

Caltech

Summary

Given $M_1(x)$ $\vdots$ $M_n(x)$ $\rightarrow$ $\sum_{i=1}^{n} \alpha_i(x) M_i(x) \approx Y(x)$

$$\mathbb{E}[M_i(x)] = 0$$

Where $\alpha_i(x) = \frac{\frac{1}{Var[M_i(x)]}}{\sum_{k=1}^{n} \frac{1}{Var[M_k(x)]}}$

1. Simplification + Gaussian ideal case
2. Directly minimize error

   Does not work

3. Assume unbiased models

Caltech

Summary

Given $M_1(x)$ $\vdots$ $M_n(x)$ $\rightarrow$ $\sum_{i=1}^{n} \alpha_i(x) M_i(x) \approx Y(x)$

$$\mathbb{E}[M_i(x)] = 0$$

Where $\alpha_i(x) = \dfrac{e^{-\lambda_i(x)}}{\sum_{k=1}^{n} e^{-\lambda_k(x)}}$

1. Simplification + Gaussian ideal case
2. Directly minimize error

   Does not work

3. Assume unbiased models
4. Learn $e^{\lambda_i(x)} \approx Var[M_i(x)]$

$$\lambda_i = \underset{l \in \mathscr{H}}{\operatorname{argmin}} \sum_{k=1}^{N} \left[e^{l(X_k)} - (Y(X_k) - M_i(X_k))^2\right]^2 + \eta \|l\|^2_{\mathscr{H}}$$

ML regression

(Neural network, Gaussian process…)

$\{X_i, Y(X_i)\}_{i=1}^{N}$

Caltech