IMPERIAL



Accumulated Aggregated *D*-optimal Designs for Estimating Main Effects in Black-Box Models

Chih-Yu Chang

Joint work with Dr. Ming-Chung Chang (Academia Sinica)

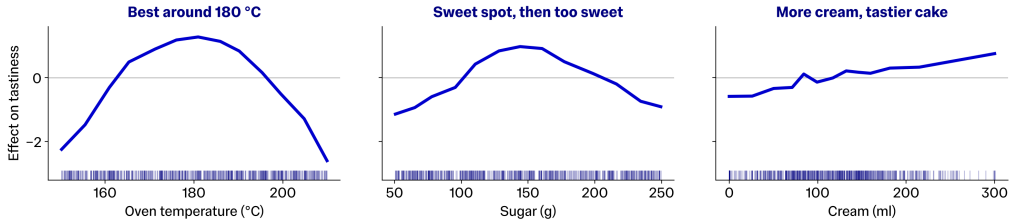
October 21, 2026

A.1 Motivation: What Makes a Cake Tasty?

Oven (°C)	Sugar (g)	Cream (ml)	Taste
188	220	257	8.5
204	201	51	5.7
164	65	63	6.2
⋮	⋮	⋮	⋮

Question: how does *each* ingredient affect the taste?

1. **Fit an ML model** (boosted trees, neural net, . . .)
2. **Plot the main effect** of each variable



A.2 Setup & Notation

- Training data $\mathcal{D}_n = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$, $\mathbf{x}_i \in \mathbb{R}^p$, $y_i \in \mathbb{R}$, where $\mathbf{X}_i = (X_{i,1}, \ldots, X_{i,p})^\top \overset{\text{i.i.d.}}{\sim} P_{\mathbf{X}}$
- Fitted prediction model $\hat{f}_n = \mathcal{A}(\mathcal{D}_n)$ from a supervised learning algorithm $\mathcal{A}$

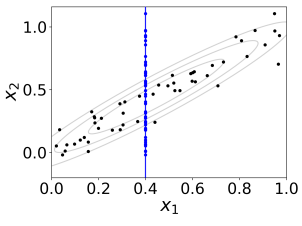
Target

We are interested in estimating the main effect of a variable on the response f using $\hat{f}_n$ and $\mathcal{D}_n$.

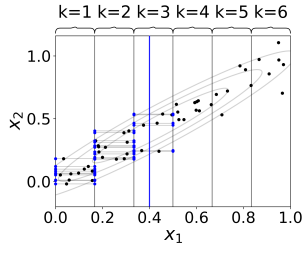
A.3 Existing Methods

- **PD** (Friedman 2001) – **fix** $X_d = t$ and average over the *marginal* of the others: $f_{d,\text{PD}}(t) = \mathbb{E}\big[f(t, X_{\setminus d})\big]$
- **ALE** (Apley & Zhu 2020) – split X_d into bins $[z_d^k, z_d^{k+1}]$ and **accumulate** *differences* up to the bin $J(t) \ni t$:

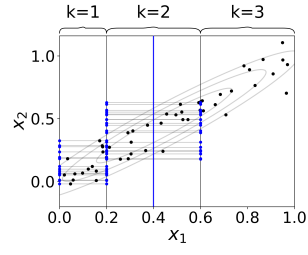
$$f_{d,\text{ALE}}(t) = \lim_{K\to\infty} \sum_{k=1}^{J(t)} \mathbb{E}\big[f(z_d^{k+1}, X_{\setminus d}) - f(z_d^k, X_{\setminus d}) \,\big|\, X_d \in [z_d^k, z_d^{k+1}]\big]$$



(a) PD



(b) ALE, $K = 6$



(c) ALE, $K = 3$

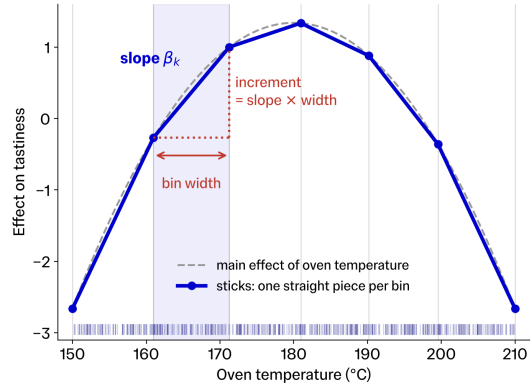
B.1 A2D2E: Definition of the Main Effect

- Split the range of X_d into K bins $[z_d^k, z_d^{k+1}]$
- In each bin, draw a **straight stick** whose slope is the **average partial derivative** $\partial f / \partial x_d$ in that bin
- The increment over the bin is slope $\times$ bin width; **stack the sticks** up to t

$$f_{d,\text{A2D2E}}(t) = \lim_{K \to \infty} \sum_{k=1}^{J(t)} \left(z_d^{k+1} - z_d^k\right) \beta_k \qquad (2)$$

$$\beta_k = \mathbb{E}\left[\frac{\partial f}{\partial x_d}(X_d, X_{\setminus d}) \,\middle|\, X_d \in [z_d^k, z_d^{k+1}] \right]$$

Cake example: oven temperature



B.2 A2D2E Estimator

$f_{d,\text{A2D2E}}(t) = \lim_{K\to\infty} \sum_{k=1}^{J(t)} \left(z_d^{k+1} - z_d^k\right) \beta_k$, where $\beta_k = \mathbb{E}\left[\frac{\partial f}{\partial x_d}(X_d, X_{\setminus d}) \,\middle|\, X_d \in [z_d^k, z_d^{k+1}]\right]$

1. The bin widths $z_d^{k+1} - z_d^k$ are known; the only unknown is the bin slope β_k
   $\Rightarrow$ estimate it by averaging over the training points in the bin: $\hat{\beta}_k = \frac{1}{|I_d^k|} \sum_{i \in I_d^k} \hat{\beta}_{i,d}$

2. **Estimate each partial derivative.** Around each training point $\mathbf{x}_i$, fit a **local linear model** $\ell_i(\mathbf{u}) = \alpha_i + \boldsymbol{\gamma}_i^\top (\mathbf{u} - \mathbf{x}_i)$ and use the coefficient of x_d
   $\Rightarrow$ the problem reduces to **estimating a slope** $\hat{\gamma}_{i,d} \approx \partial f / \partial x_d(\mathbf{x}_i) = \beta_{i,d}$

B.3 Local Linear Regression

Fit a **local linear model** on a neighbourhood $\mathcal{S}_i$ of $\mathbf{x}_i$:

$$\ell_i(\mathbf{u}) = \alpha_i + \boldsymbol{\gamma}_i^\top (\mathbf{u} - \mathbf{x}_i), \qquad \mathbf{u} \in \mathcal{S}_i, \tag{4}$$

and take $\hat{\gamma}_{i,d}$, the d-th component of the fitted slope vector, as the estimate of $\partial f / \partial x_d(\mathbf{x}_i)$.

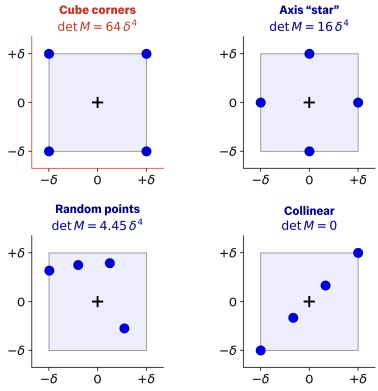
Lemma 1 (local linear approximation of f)

Let $\mathcal{S}_i(\delta) = \{\mathbf{u} \in \mathbb{R}^p : \|\mathbf{u} - \mathbf{x}_i\|_\infty \le \delta\}$ and $f \in C^2$ near $\mathbf{x}_i$. If ℓ_i minimises $\sup_{\mathbf{u} \in \mathcal{S}_i(\delta)} |f(\mathbf{u}) - \ell_i(\mathbf{u})|$, then for each $d \in [p]$,

$$\gamma_{i,d} \longrightarrow \frac{\partial f}{\partial x_d}(\mathbf{x}_i) \qquad \text{as } \delta \to 0.$$

Good news: $\hat{f}_n$ is free to query. **Question:** which points in $\mathcal{S}_i$ give the best estimate of γ_i? $\rightarrow$ **step 3**.

B.4 *D*-optimal Design: Cube Corners



4 queries around $\mathbf{x}_i$ (+), model $\alpha + \gamma_1 u_1 + \gamma_2 u_2$

Where to query $\hat{f}_n$ around $\mathbf{x}_i$?

Information matrix of the design points $\mathbf{u}_{i,r}$:

$$M_i = \sum_r \begin{pmatrix} 1 \\ \mathbf{u}_{i,r} - \mathbf{x}_i \end{pmatrix} \begin{pmatrix} 1, & \mathbf{u}_{i,r} - \mathbf{x}_i \end{pmatrix}$$

- $\mathrm{Cov}(\hat{\gamma}_i) \propto M_i^{-1} \Rightarrow$ ***D*-optimal: maximise** $\det M_i$
- $p = 2$: **corners win**, $\det M_i = 64\,\delta^4$
- Any p: the **cube vertices**

$$\left\{ \mathbf{x}_i + \delta\varepsilon : \ \varepsilon \in \{\pm 1\}^p \right\}$$

B.5 Closed Form

On the vertices, $\mathbf{u} - \mathbf{x}_i = \delta\varepsilon$, and the sign vectors satisfy

$$\sum_{\varepsilon} \varepsilon_d = 0, \qquad \sum_{\varepsilon} \varepsilon_d \varepsilon_j = 2^p \, \mathbf{1}\{j = d\}$$

so the design is **orthogonal**:

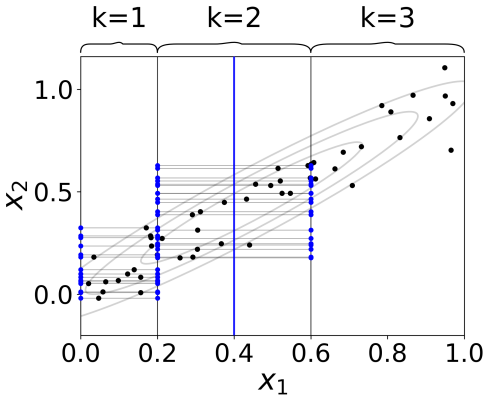
$$M_i = 2^p \, \mathrm{diag}(1, \delta^2, \ldots, \delta^2)$$

Inverting a diagonal matrix is trivial, so the local slope is a **signed average** of the 2^p queries:

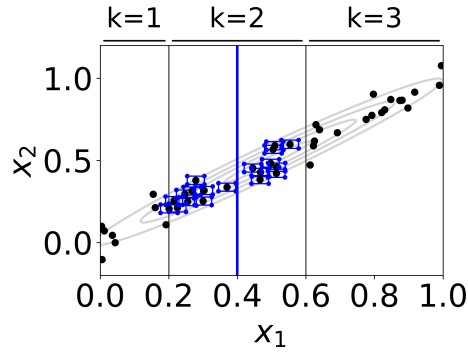
$$\hat{\gamma}_{i,d} = \frac{1}{2^p \delta} \sum_{\varepsilon \in \{-1,+1\}^p} \varepsilon_d \, \hat{f}_n(\mathbf{x}_i + \delta\varepsilon)$$

- **No matrix inversion**, and the same 2^p queries give the slope for **every** d
- $2\delta \, \hat{\gamma}_{i,d}$ is the textbook **main effect of factor** d in a 2^p factorial experiment on $\hat{f}_n$

B.6 Why A2D2E is Less Sensitive to OOD Evaluations



ALE, $K = 3$: endpoints up to h_k away



A2D2E, $K = 3$: vertices exactly δ away

C.1 Consistency of A2D2E

Theorem (informal)

Suppose f is smooth, $\hat{f}_n \to f$ uniformly, and as $n \to \infty$: bins get finer ($K_n \to \infty$), the cube shrinks but never exceeds a bin ($\delta_n \leq h_{K_n}$), and $\|\hat{f}_n - f\|_\infty / \delta_n \to 0$. Then $\sup_t \left|\hat{f}_{d,A2D2E}(t) - f_{d,A2D2E}(t)\right| \to 0$ almost surely.

The error splits into four terms:

Error term	Size
(A) surrogate error	$\|\hat{f}_n - f\|_\infty / \delta_n$
(B) sampling error	$\max_k \lvert\tilde{\beta}_k - \beta_k\rvert$
(C) hypercube bias	$o(\delta_n)$
(D) bin bias	$\to 0$ as $K_n \to \infty$

D.1 Simulation Setup & Main Results

Setup

- 6 **additive** benchmarks f_0–f_5, $p = 2$ to 8: all methods share the same true main effect
- Correlation: **independent**, **low**, **high**
- Surrogate: $\mathcal{K}$-NN (sensitive to OOD inputs)
- $K = 40$ quantile bins, δ = bin width
- Metric: ORMSE vs truth, mean of 100 reps

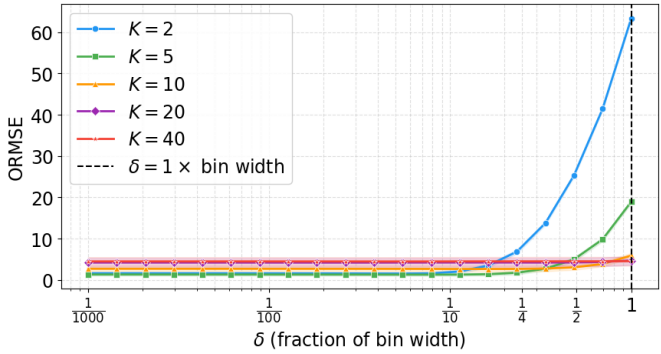
Findings

- **Independent:** PD wins – no extrapolation
- **Any correlation:** A2D2E is best or second, and the gap grows with dependence
- **Beats ALE in all 18 rows** – same target and bins; only *where* $\hat{f}_n$ is queried differs

Table 2: mean ORMSE, $\mathcal{K}$-NN. **Bold** = best, underline = second

Dep.	f	PD	M	ALE	DALE	A2D2E
Indep.	f_0	**.0306**	.1299	.1093	.1953	.0886
	f_1	**.0637**	.2021	.1448	.4040	.1336
	f_2	**.2125**	.3421	.3990	.4835	.3704
	f_3	**.0925**	.3056	.2928	.5024	.2638
	f_4	**.3578**	1.4528	1.1314	1.5716	.8381
	f_5	**.2014**	.2350	.2499	.2666	.2397
Low	f_0	.0932	.3639	.1123	.2006	**.0834**
	f_1	.2306	.3693	.1728	.4147	**.1504**
	f_2	**.2930**	.6823	.3514	.4833	.3113
	f_3	**.1720**	1.3514	.2922	.5025	.1998
	f_4	1.7226	4.1034	1.4658	1.5712	**1.3270**
	f_5	.2508	.8672	.2487	.2664	**.2266**
High	f_0	.1161	.3789	.1220	.1985	**.0907**
	f_1	.2768	.3893	.2111	.4132	**.1956**
	f_2	.3516	.7958	.3367	.4823	**.2967**
	f_3	**.2038**	1.4248	.2821	.4977	.2060
	f_4	1.9022	4.3050	1.6043	1.5721	**1.5175**
	f_5	.2538	1.0303	.2451	.2660	**.2217**

D.2 Robustness to K and δ



ORMSE over 100 reps, one curve per number of bins K; x-axis: δ as a fraction of the bin width

Adversarial OOD test (Gkolemis et al. 2023): f explodes off the data manifold, so any far-away query is heavily penalised

Practical default: $K = 40$, $\delta \leq$ bin width – no tuning needed.

D.3 Wall-Clock Experiments

$f = x_1 + x_2^2$, 1-NN surrogate, $K = 40$, δ = bin width; seconds, mean of 5 runs

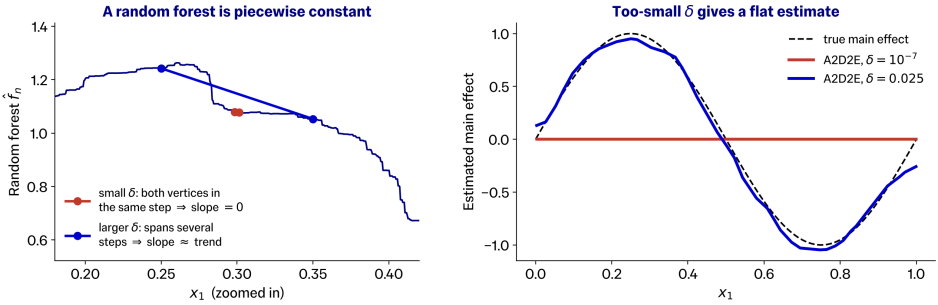
Single main effect, $p = 4$, varying n

n	PD	ALE	A2D2E
250	0.005	0.001	0.002
500	0.010	0.001	0.003
1000	0.024	0.001	0.007
2000	0.059	0.003	0.015
4000	0.137	0.005	0.035

All p main effects, $n = 500$, varying p

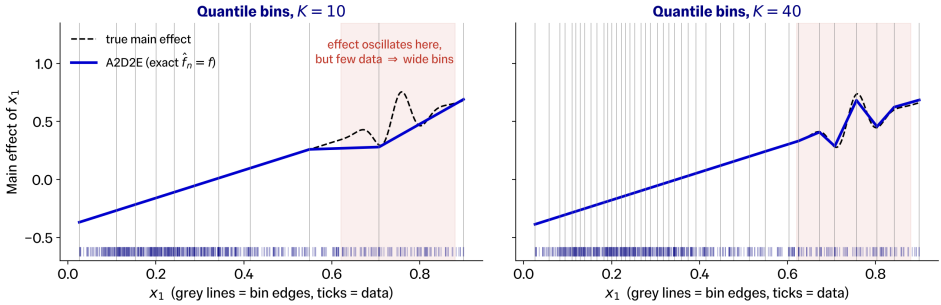
p	PD	ALE	A2D2E
2	0.050	0.003	0.004
3	0.119	0.007	0.007
4	0.238	0.009	0.016
5	0.423	0.012	0.034
8	1.476	0.023	0.313

E.1 Tailoring A2D2E to the Fitted Model



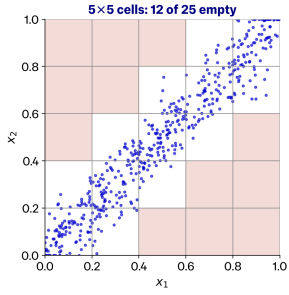
- A2D2E treats $\hat{f}_n$ as a generic black box, but the right δ **depends on what $\hat{f}_n$ is**
- A **random forest** is piecewise constant: if δ is smaller than its steps, all vertices land in the same leaf $\Rightarrow$ every local slope is 0 $\Rightarrow$ the main effect is **flat**

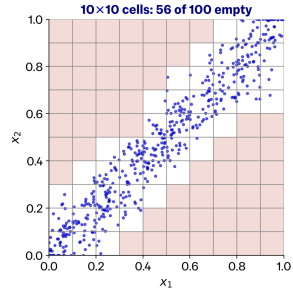
E.2 Choosing the Partition



- ALE, DALE and A2D2E all fit a **local linear** piece in each bin, and bins come from **quantiles** – the same number of data points per bin
- Where the effect oscillates but data are sparse, bins are wide and the **wiggles are averaged away** (left); more bins everywhere (right) leaves fewer points per bin and noisier slopes

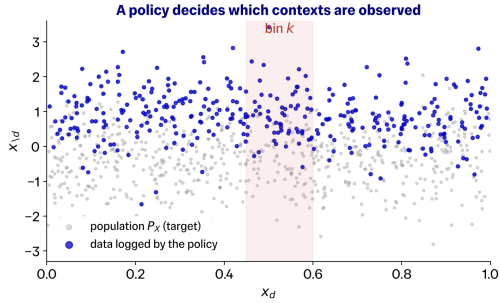
E.3 Higher-Order Effects

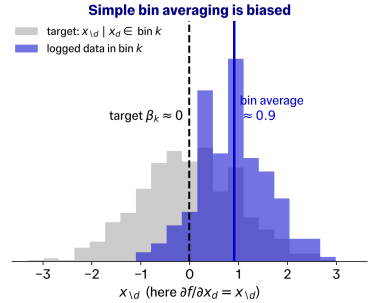




- The same recipe extends to **joint effects**, e.g. $f_{12}(x_1, x_2)$: partition into **cells**, estimate local effects, accumulate
- Under correlation, many cells contain **no training data** – 56 of 100 here, and far worse as the dimension grows
- Current fixes are **heuristic** (merge, skip or extrapolate into empty cells)

E.4 Explainability under Non-i.i.d. Data





- We estimate β_k by **averaging over the data in the bin** – valid only for i.i.d. samples from P_X
- Data collected by a **policy** (bandits, RL, adaptive experiments) over-represent what the policy prefers $\Rightarrow$ **biased**
- **Open:** recover the right conditional distribution from adaptively collected data, with guarantees

E.5 Benchmarking Main-Effect Estimators

To compare methods, we need a **ground-truth main effect** – and that is hard for two reasons:

1. **The target is not the same across methods**
2. **The ground truth is hard to compute when f is not additive**

Open: non-additive benchmarks with **tractable targets**, and evaluation criteria that are fair across targets.

Thank you.
Questions?

Backup: Computational Complexity

C = cost of one query of $\hat{f}_n$; single variable d

- **PD:** $\mathcal{O}(KnC)$ – for each of the K grid points, query $\hat{f}_n$ once per observation
- **ALE:** $\mathcal{O}(2nC)$ – two bin endpoints per observation
- **A2D2E:** formulated as local regression, but the orthogonal vertex design gives the closed form (6) – **no matrix inversion**; cost per observation $\mathcal{O}(2^p C)$, total $\mathcal{O}(2^p nC)$

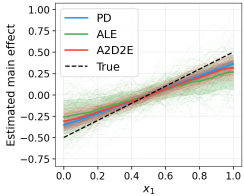
All p main effects jointly:

- The 2^p vertex evaluations $\{\hat{f}_n(\mathbf{z}_{i,r})\}_r$ **do not depend on the target dimension** d
- Compute them once per $\mathbf{x}_i$, then read off $\hat{\gamma}_{i,d}$ for every $d \in [p]$ at negligible extra cost

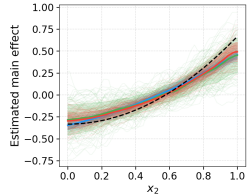
Method	**One dimension**	**All p dimensions**
PD	$\mathcal{O}(KnC)$	$\mathcal{O}(pKnC)$
ALE	$\mathcal{O}(2nC)$	$\mathcal{O}(2pnC)$
A2D2E	$\mathcal{O}(2^p nC)$	$\mathcal{O}(2^p nC)$ (no extra factor p)

Backup: Estimated Main Effects: $f_0 = x_1 + x_2^2 + 0 \cdot x_3$

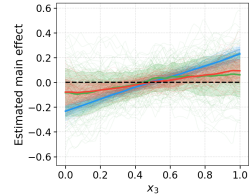
Low dependence, $\mathcal{K}$-NN; PD (blue), ALE (green), A2D2E (red), truth (dashed)



(a) x_1



(b) x_2



(c) x_3 (null)

Solid = mean over replications; shaded = 95% CI; faint lines = individual replications.

- All three recover the **active** effects of x_1 and x_2 reasonably well; ALE shows the largest variability across replications
- The **null variable** x_3 is most revealing: PD shows a **spurious linear trend** – x_3 is correlated with x_1, x_2 and PD extrapolates
- ALE reduces the trend but keeps wide replication bands; A2D2E stays **closest to the true zero effect** with the tightest uncertainty

Backup: Algorithm 1 (A2D2E)

Require: $\hat{f}_n$, $\mathcal{D}_n$, number of bins K, cube half-width δ, target variable index d

1. Let $\{z_d^k\}_{k=1}^{K+1}$ be bin boundaries on axis d (quantiles of X_d, so bins have $\approx$ equal counts)
2. Initialise $\hat{f}_d(z_d^1) \leftarrow 0$
3. **for** $k = 1$ **to** K **do**

   Initialise $S_k \leftarrow 0$

   for $i \in I_d^k$ **do** ▷ **Steps 3 + 2: design, then local slope**

   Construct $\mathcal{V}_i(\delta) = \{\mathbf{z}_{i,1}, \ldots, \mathbf{z}_{i,2^p}\} = \{\mathbf{x}_i + \delta\boldsymbol{\varepsilon} : \boldsymbol{\varepsilon} \in \{-1, +1\}^p\}$

   Evaluate $y_{i,r} \leftarrow \hat{f}_n(\mathbf{z}_{i,r})$ for $r = 1, \ldots, 2^p$

   Compute $\hat{\gamma}_i$ (closed form (6)) and set $S_k \leftarrow S_k + \hat{\gamma}_{i,d}$

   end for

   Averaged bin-level slope $\hat{\beta}_k \leftarrow S_k / |I_d^k|$ ▷ **Step 1: aggregate**

   $\hat{f}_d(z_d^{k+1}) \leftarrow \hat{f}_d(z_d^k) + (z_d^{k+1} - z_d^k)\,\hat{\beta}_k$ ▷ **Step 1: accumulate**
4. **end for**
5. **return** $\hat{f}_d(z_d^k) - \frac{1}{K}\sum_{k'=1}^{K} \hat{f}_d(z_d^{k'+1})$ for all $k = 1, \ldots, K+1$

Backup: Semi-Synthetic Study: Building a Realistic Ground Truth

`scikit-learn` diabetes: $n = 442$, $p = 10$; response = disease progression

Covariates: age, sex, `bmi`, `bp`, six serum measurements `s1`–`s6`.

Goal: a smooth *additive* truth that looks like real data, so all estimands coincide and ORMSE is meaningful.

Construction

1. For each covariate x_d, build a dictionary of univariate bases:
$$x_d^j \ (j = 1, \ldots, 5), \qquad \sin(2\pi k x_d),\ \cos(2\pi k x_d) \ (k = 1, \ldots, 5)$$
$\Rightarrow$ 15 bases per covariate, **150 candidate additive terms**
2. Forward selection with 5-fold CV to select a sparse additive model
$$f(\mathbf{x}) = \tau_0 + \sum_{m \in \hat{S}} \tau_m \phi_m(\mathbf{x}), \qquad \text{each } \phi_m \text{ depends on one covariate}$$
3. Final model: **20 additive terms**, CV $R^2 = 0.5371$
4. Use this fitted model as the ground-truth f in all simulations

Backup: Semi-Synthetic Study: Data Generation and Models

Each Monte Carlo replication ($n = 200$):

- Latent $Z_i \sim \mathrm{Unif}(0, 1)$; features $X_{i,d}^{\text{raw}} = Z_i + \epsilon_{i,d},\ \epsilon_{i,d} \sim \mathcal{N}(0, \sigma^2)$ – all features share $Z_i \Rightarrow$ dependence controlled by σ
- Rescale each feature to the empirical support of the corresponding diabetes covariate
- $Y_i = f(\mathbf{X}_i) + \eta_i$, with Gaussian η_i at 10% of the signal s.d.

Two fitted models at two dependence levels:

Model	Dependence	Details
Neural net	$\sigma = 0.15$	MLP, one hidden layer of 16 ReLU units, L-BFGS, ℓ_2 penalty 10^{-4}, ≤ 200 iterations
$\mathcal{K}$-NN	$\sigma = 0.03$ (higher)	neighbours and weighting chosen by 5-fold CV in every replication

Centred main effects from PD, ALE, DALE, A2D2E; $K = 50$ bins for ALE/DALE/A2D2E; $\delta = 10\%$ of bin width; **200 replications**.

Backup: Semi-Synthetic Study: Results (Table 4)

Mean ORMSE with 95% CI over 200 replications; **bold** = best per fitted model

$\mathcal{A}$	PD	ALE	DALE	A2D2E
NN	10.8424 ± 0.2307	7.4320 ± 0.1152	7.0262 ± 0.1040	7.0226 ± 0.1042
$\mathcal{K}$-NN	15.3968 ± 0.0458	15.1702 ± 0.1795	15.2786 ± 0.0481	14.7760 ± 0.0396

Neural network ($\sigma = 0.15$)

- A2D2E **substantially improves** over PD (10.84 $\rightarrow$ 7.02) and ALE (7.43 $\rightarrow$ 7.02)
- Matches DALE – expected, since DALE uses the exact gradient of the smooth NN, information the other methods never see

$\mathcal{K}$**-NN** ($\sigma = 0.03$, higher dependence)

- A2D2E improves over **all three** baselines; paired differences relative to A2D2E: **0.6208** (PD), **0.3942** (ALE), **0.5026** (DALE)

$\Rightarrow$ A2D2E is especially useful when the fitted model is **locally irregular** and the covariates are **highly dependent**

Backup: Practical Guidance: Which Method When?

Situation	Prefer	Why
Features (nearly) independent	PD	no extrapolation bias; averages over all n points
Correlated features **and** OOD-sensitive surrogate ($\mathcal{K}$-NN, local GP, trees)	**A2D2E**	queries stay within δ of the data; largest gains observed
Correlated features, smooth well-calibrated surrogate	ALE or A2D2E	gains of A2D2E are modest; ALE is cheaper
Gradients of $\hat{f}_n$ cheaply available and reliable	DALE or A2D2E	similar accuracy; A2D2E needs no gradients
Non-differentiable / pure black-box $\hat{f}_n$	**A2D2E**	only function evaluations needed
Large p	ALE, or A2D2E with a fractional design	full factorial costs 2^p queries per point

Guidance synthesised from the paper's experiments; a principled selection rule is future work.

Backup: Limitations

Limitations

- Advantage is most pronounced when the surrogate is OOD-sensitive – from the **data-sampling process** (correlation) or the **surrogate itself**. When neither is present, gains over ALE are modest and ALE is computationally cheaper
- Full factorial design: 2^p queries per point; the exponential factor is visible by $p = 8$
- Benchmarking non-additive main effects remains hard: the target depends on the joint covariate distribution

Backup: Summary

1. **Problem.** Main-effect estimators query $\hat{f}_n$ off the data: PD extrapolates, ALE makes OOD evaluations, DALE needs gradients
2. **Insight.** All methods differ only in *where* they evaluate $\hat{f}_n \Rightarrow$ main-effect estimation is an **experimental-design problem**
3. **Method.** A2D2E: around each training point, a D-optimal 2^p hypercube design gives a closed-form local slope; aggregate within bins, accumulate across bins
4. **Theory.** Consistent for the ALE estimand with only a surrogate $\hat{f}_n$; explicit four-term error bound; hypercube bias is $o(\delta)$
5. **Computation.** No matrix inversion; 2.5–15$\times$ faster than PD; evaluations shared across dimensions
6. **Evidence.** Best or second-best in every correlated setting; most stable under surrogate misspecification; insensitive to K and δ

Backup: What Needs to be Controlled?

The **target** $f_{d,\text{A2D2E}}$ is a $K \to \infty$ limit under the true f. The **estimator** uses finite K, finite n, and the fitted surrogate $\hat{f}_n$ in place of f.

Three distinct sources of error:

1. **Statistical / surrogate error** – estimating slopes from $\hat{f}_n$ and n observations rather than from f
2. **Local-linear (hypercube) bias** – approximating the true derivative $\partial f / \partial x_d(\mathbf{x}_i)$ by a finite-difference slope over a cube of fixed half-width δ
3. **Bin-discretisation bias** – truncating the population definition at a finite number of bins K

Result

All three vanish simultaneously under mild conditions that relate the growth of K, the shrinkage of δ, and the convergence rate of $\hat{f}_n$.

Backup: Assumptions

(A1) Data $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ i.i.d., and the domain $\mathcal{X}$ is compact

(A2) Response $f \in C^2(\mathcal{X})$, $\partial f/\partial x_d$ continuous; marginal density p_d of X_d has compact support S_d with $p_d(z) > 0$ on S_d

(A3) Surrogate $\sup_{\mathbf{x}\in\mathcal{X}} |\hat{f}_n(\mathbf{x})| < \infty$ and $\|\hat{f}_n - f\|_{L^\infty} := \sup_{\mathbf{x}\in\mathcal{X}} |\hat{f}_n(\mathbf{x}) - f(\mathbf{x})| \to 0$ a.s.

(A4) Design & partition rates $K = K_n \to \infty$, $\delta = \delta_n > 0$, all $\mathbf{z}_{i,r} = \mathbf{x}_i + \delta_n \boldsymbol{\varepsilon} \in \mathcal{X}$, and

$$h_{K_n} := \max_k \left(z_d^{k+1} - z_d^k\right) \to 0, \qquad \delta_n \le h_{K_n}, \qquad \frac{\|\hat{f}_n - f\|_{L^\infty}}{\delta_n} \to 0 \text{ a.s.}$$

(A5) Uniform SLLN over bins Empirical bin averages satisfy a uniform strong law; e.g. for quantile bins it suffices that $K_n \log n/n \to 0$

Interpretation of (A4):

- bins get finer, and the cube never exceeds the bin width
- δ_n must shrink **slower** than the sup-norm error of $\hat{f}_n$: the slope divides by δ, so surrogate error is amplified by $1/\delta$

Backup: Theorem 1: Consistency of A2D2E

Theorem

Under (A1)–(A5), define $g_{K_n}(t) := \sum_{k=1}^{J(t)} (z_d^{k+1} - z_d^k)\beta_k$ *with* $\beta_k := \mathbb{E}[\partial f/\partial x_d(\mathbf{X}) \mid X_d \in [z_d^k, z_d^{k+1}]]$*, the exact finite-*$K_n$ *bin average. Then for* $t \in [a_d, b_d]$*,*

$$\sup_t \left|\hat{f}_{d,A2D2E}(t) - f_{d,A2D2E}(t)\right| \le \underbrace{(b_d - a_d)\frac{\|\hat{f}_n - f\|_{L^\infty}}{\delta_n}}_{\text{(A) surrogate error}} + \underbrace{(b_d - a_d)\max_k \left|\tilde{\beta}_k^{(n)} - \beta_k^{\delta_n}\right|}_{\text{(B) sampling error}}$$
$$+ \underbrace{(b_d - a_d)\,O(\delta_n)}_{\text{(C) hypercube bias}} + \underbrace{\sup_t \left|g_{K_n}(t) - f_{d,A2D2E}(t)\right|}_{\text{(D) bin-discretisation bias}},$$

and each term $\to 0$ *a.s. as* $n \to \infty$*. Hence, for each fixed* t*,* $\hat{f}_{d,A2D2E}(t) \to f_{d,A2D2E}(t)$ *a.s.*

- Consistent for the **same population target as ALE**, even though only the fitted surrogate $\hat{f}_n$ is ever queried
- The bound is explicit: each term corresponds to one of the three error sources (plus sampling)

Backup: Evaluation Protocol

Why additive benchmarks? Ground truth is unavailable in real data, so synthetic benchmarks are standard. For additive $f(\mathbf{x}) = \sum_d h_d(x_d)$, the partial derivative $\frac{\partial f}{\partial x_d} = h_d'(x_d)$ depends only on x_d, so

$$f_{d,\mathrm{ALE}}(t) = \int_{a_d}^{t} \mathbb{E}\Big[\tfrac{\partial f}{\partial x_d}(z, X_{\setminus d}) \,\Big|\, X_d = z\Big]\, dz = \int_{a_d}^{t} h_d'(z)\, dz = h_d(t) - h_d(a_d),$$

identical to $f_{d,\mathrm{PD}}(t) = h_d(t) + \sum_{j \neq d} \mathbb{E}[h_j(X_j)] - c$ up to a constant. $\Rightarrow$ PD, ALE, DALE and A2D2E share **the same estimand regardless of dependence** – a fair comparison.

Centring: all estimates and the truth are centred to mean zero over the evaluation points.

Metric: Overall RMSE over a grid of 100 equally spaced points per dimension,

$$\mathrm{ORMSE} = \frac{1}{p} \sum_{d=1}^{p} \sqrt{\mathrm{MSE}_d},$$

averaged over 100 independent replications (95% CIs reported).

Defaults: $K = 40$ (Apley & Zhu 2020), quantile bins, δ = bin width (0.025). DALE uses the exact gradient, or finite differences when $\hat{f}_n$ is not differentiable.

Backup: Benchmark Functions and Data Generation

Table 1: additive benchmarks on $[0,1]^p$, $u_d = 2x_d - 1$, $\sigma(z) = 1/(1+e^{-z})$

f (p)	Definition
f_0 (3)	$x_1 + x_2^2 + 0 \cdot x_3$
f_1 (2)	$\sin(u_1) + u_2^2$
f_2 (4)	$\sin(10u_1) + \sin(u_2) + (u_3^3 - u_3) + \sigma(10u_4)$
f_3 (4)	$\mathbf{1}(u_1>0)u_1^2 + \lvert u_2\rvert^{1/2}\mathrm{sgn}(u_2)$ $+ \sin(\pi u_3/2) + u_4 \log(\lvert u_4\rvert + 1)$
f_4 (4)	$10\sin(u_1) + 0.1u_2^2 + 5e^{-u_3^2} + 0.05u_4$
f_5 (8)	$\sum_{k=1}^{8}\left[\frac{1}{k}\sin(k\pi u_k) + k\cos(\frac{\pi u_k}{2k})\right]$

Spans functional forms, smoothness levels (incl. kinks and discontinuous derivatives in f_3), null variables (f_0) and dimensions $p = 2$ to 8.

Data: $n = 100p$ points; mean-zero noise with variance 30% of the signal variance ($R^2 \approx 0.8$)

$x_1 \sim \mathrm{Unif}(0,1)$; remaining variables at three dependence levels:

- **Independent:** all $\sim \mathrm{Unif}(0,1)$
- **Low:** $x_d = x_1 + \varepsilon_d$, $\varepsilon_d \sim \mathcal{N}(0, 0.1^2)$
- **High:** $x_d = x_1 + \varepsilon_d$, $\varepsilon_d \sim \mathcal{N}(0, 0.05^2)$

Surrogate: $\mathcal{K}$-NN regression (James et al. 2013)

- fast, interpretable, but prediction is an average of $\mathcal{K}$ neighbours $\Rightarrow$ **sensitive to OOD inputs**
- tuned per function by 10-fold CV on $500p$ uniform points; $\mathcal{K} \in \{1, \ldots, \min(50, 10p)\}$, uniform or distance weights

Backup: Results: $\mathcal{K}$-NN Regression (Table 2)

Mean ORMSE, 100 reps. **Bold** = best, underline = second (CIs in backup)

Dep.	f	PD	M	ALE	DALE	A2D2E
Indep.	f_0	**.0306**	.1299	.1093	.1953	.0886
	f_1	**.0637**	.2021	.1448	.4040	.1336
	f_2	**.2125**	.3421	.3990	.4835	.3704
	f_3	**.0925**	.3056	.2928	.5024	.2638
	f_4	**.3578**	1.4528	1.1314	1.5716	.8381
	f_5	**.2014**	.2350	.2499	.2666	.2397
Low	f_0	.0932	.3639	.1123	.2006	**.0834**
	f_1	.2306	.3693	.1728	.4147	**.1504**
	f_2	**.2930**	.6823	.3514	.4833	.3113
	f_3	**.1720**	1.3514	.2922	.5025	.1998
	f_4	1.7226	4.1034	1.4658	1.5712	**1.3270**
	f_5	.2508	.8672	.2487	.2664	**.2266**
High	f_0	.1161	.3789	.1220	.1985	**.0907**
	f_1	.2768	.3893	.2111	.4132	**.1956**
	f_2	.3516	.7958	.3367	.4823	**.2967**
	f_3	**.2038**	1.4248	.2821	.4977	.2060
	f_4	1.9022	4.3050	1.6043	1.5721	**1.5175**
	f_5	.2538	1.0303	.2451	.2660	**.2217**

A2D2E's rank, by dependence

	best	2nd
Independent	0/6	4/6
Low	4/6	2/6
High	5/6	1/6

- A2D2E is **never worse than second** under any correlation
- Beats ALE and DALE in **all 18** rows

Backup: Reading Table 2: Three Insights

1. **PD wins under independence** – as expected: with independent features the artificial points $(t, \mathbf{x}_{i,\setminus d})$ are in-distribution, so there is no extrapolation bias, and PD averages over all n points
2. **A2D2E overtakes PD already at low dependence** for most functions, and the advantage **widens** under high dependence
   - even mild correlation is enough to expose PD's extrapolation
   - e.g. f_4, high dependence: PD 1.90 vs A2D2E 1.52
3. **DALE performs poorly throughout**, despite its extra requirement of gradient access – $\mathcal{K}$-NN is piecewise constant, so its derivative (via finite differences) is unstable

Versus ALE (same estimand, same bins)

A2D2E has lower ORMSE than ALE in **all 18 rows**, including under independence, with the largest gains for high-amplitude functions (f_4 independent: 1.13 $\rightarrow$ 0.84).

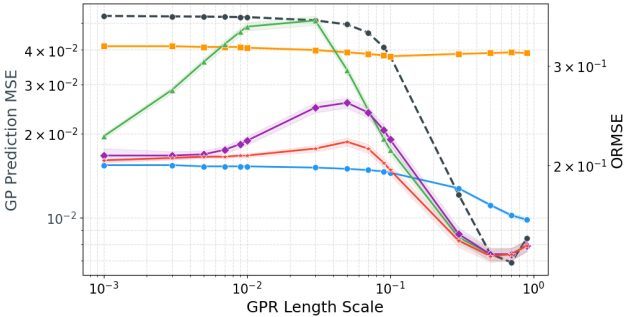
Backup: Case Study: Gaussian Process Regression – Setup

- GPR is a non-parametric predictor governed by its kernel; for the RBF kernel, the **length scale** ℓ controls smoothness
  - small ℓ: highly local, wiggly fit – interpolates training data but **generalises poorly to unseen regions**
  - large ℓ: oversmoothed surface that may miss local structure
- This directly affects any method that evaluates the surrogate at **OOD inputs** – a controlled way to vary OOD sensitivity

Design of the study

- Locally sharp function: $f(x_1, x_2, x_3) = \exp\big(-100(x_1 - 0.5)^2\big) + x_2 + 0 \cdot x_3$
- High-dependence sampling, all other settings as before
- GPR with RBF kernel fitted at **20 fixed length scales** from highly local (10^{-3}) to oversmoothed (≈ 1)
- 500 replications per length scale

Backup: Case Study: Gaussian Process Regression – Results



Black dashed: GP prediction MSE (left axis). ORMSE (right axis): PD (blue), M (orange), ALE (green), DALE (purple), A2D2E (red); bands = 95% CI.

Two strengths of A2D2E

1. **Most stable ALE-type method** across the entire length-scale range
   - ALE deteriorates sharply at moderate $\ell \in [0.03, 0.05]$
   - DALE has high variance throughout the intermediate regime
   - A2D2E keeps consistently low ORMSE regardless of GP misspecification
2. At the well-calibrated region $\ell \in [0.1, 0.3]$, where GP prediction MSE drops sharply, A2D2E is **competitive with PD**

Message: when the surrogate is unreliable off the data, *where* you query it matters most.

Backup: Sensitivity Analysis – Setup

Following Gkolemis et al. (2023): a severely adversarial OOD setting

$$f(\mathbf{x}) = x_1 x_2 + x_1 x_3 - a\big((x_1 - x_2)^2 - \tau^2\big)\mathbf{1}[x_1 - x_2 \geq \tau] + a\big((x_1 - x_2)^2 - \tau^2\big)\mathbf{1}[x_2 - x_1 \geq \tau],$$

with $\tau = 1.2$, $a = 7$: **mild** inside $|x_1 - x_2| < \tau$, but **diverges rapidly** outside – any query off the data manifold is heavily penalised.

Data ($n = 1000$):

- x_1 clustered around five centres $\{1.5, 3.0, 5.0, 6.3, 8.2\}$ (s.d. 0.1), plus a few boundary points; range $[0, 10]$
- $x_2 = x_1 + \varepsilon_2$, $\varepsilon_2 \sim \mathcal{N}(0, 0.6^2)$ – strongly correlated with x_1
- $x_3 \sim \mathcal{N}(0, 20^2)$ – independent noise variable

No surrogate: as in Gkolemis et al., the true f is queried directly, isolating the effect of evaluation locations.

Ground truth for x_1: in the mild region $\partial f / \partial x_1 = x_2 + x_3$, and $\mathbb{E}[X_2 + X_3 \mid X_1 = z] = z$, so the local effect is z; outside it, the penalty adds curvature and the truth is computed by Monte Carlo integration over $X_{\setminus 1} \mid X_1 = z$.

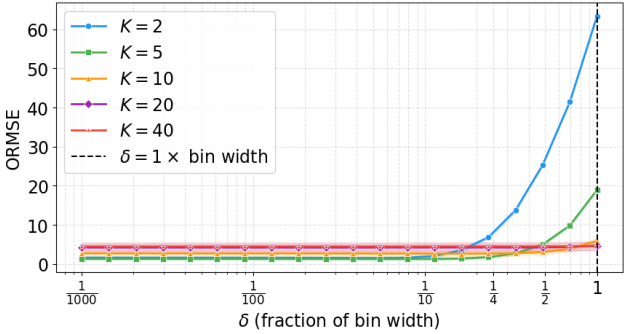
Backup: Sensitivity to the Number of Bins K (Table 3)

Mean ORMSE ($\pm$95% CI) over 100 reps; $\delta = 1/40 = 0.025$ fixed across K

	Number of bins K							
Method	1	2	3	4	5	10	20	40
ALE	79.97 ± 0.36	37.96 ± 0.32	21.06 ± 0.28	15.14 ± 0.31	10.03 ± 0.24	4.64 ± 0.23	4.92 ± 0.36	5.00 ± 0.34
DALE	3.15 ± 0.14	1.81 ± 0.17	1.86 ± 0.16	1.68 ± 0.17	1.62 ± 0.16	3.09 ± 0.22	4.76 ± 0.35	5.01 ± 0.34
A2D2E	3.14 ± 0.14	1.80 ± 0.17	1.85 ± 0.16	1.67 ± 0.17	1.60 ± 0.16	3.08 ± 0.22	4.75 ± 0.35	4.99 ± 0.34

- A2D2E has the **lowest mean ORMSE for every** K
- **ALE collapses at small** K: with $K = 1$ its endpoints span the whole range $[0, 10]$ and land in the divergent region (ORMSE 80 vs 3)
- A2D2E is essentially **indistinguishable from DALE** (differences well within the CIs) – but **without gradient access**
- With δ fixed, A2D2E's queries stay local however coarse the bins are – locality is decoupled from K
- Error grows again for large K for all three methods: fewer points per bin, noisier $\hat{\beta}_k$

Backup: Sensitivity to the Cube Half-Width δ



ORMSE over 100 replications for $K \in \{2, 5, 10, 20, 40\}$; x-axis: δ as a fraction of the bin width.

- Fix K and sweep δ from $\frac{1}{1000}$ to $1\times$ the bin width
- For $K \geq 10$: **flat** across three orders of magnitude of δ
- Only for very coarse bins ($K = 2, 5$) does error grow, and only as δ approaches the full bin width – then the vertices themselves reach the divergent region

Practical default

A2D2E is **not sensitive** to its hyperparameters: $K = 40$ with $\delta \leq$ bin width is a safe choice; smaller δ is safer when bins are coarse.