Random Gradient-Free Optimization in Infinite Dimensional Spaces

Daniel Csillag
FGV EMAp

June 17, 2026

Joint work with Caio Peixoto, Bernardo Costa and Yuri Saporito

Functional Optimization

$$\underset{f\,:\,\mathcal{X}\to\mathcal{Y}}{\operatorname{argmin}}\,\mathcal{R}(f)$$

Functional Optimization

$$\operatorname*{argmin}_{f\,:\,\mathcal{X}\to\mathcal{Y}} \mathcal{R}(f)$$

$$\operatorname*{argmin}_{\theta\in\mathbb{R}^d} F(\theta), \qquad F(\theta) = \mathcal{R}(f_\theta)$$

Functional Optimization

$$\operatorname*{argmin}_{f\,:\,\mathcal{X}\to\mathcal{Y}} \mathcal{R}(f)$$

$$\operatorname*{argmin}_{\theta\in\mathbb{R}^d} F(\theta), \qquad F(\theta) = \mathcal{R}(f_\theta)$$

Loss & gradient operator:

$$F : \mathbb{R}^d \to \mathbb{R}, \qquad \nabla F : \mathbb{R}^d \to \mathbb{R}^d$$

Functional Optimization

$$\operatorname*{argmin}_{f\,:\,\mathcal{X}\to\mathcal{Y}} \mathcal{R}(f)$$

$$\operatorname*{argmin}_{\theta\in\mathbb{R}^d} F(\theta), \qquad F(\theta) = \mathcal{R}(f_\theta)$$

Loss & gradient operator:

$$F : \mathbb{R}^d \to \mathbb{R}, \qquad \nabla F : \mathbb{R}^d \to \mathbb{R}^d$$

Optimize with gradient descent:

$$\theta_{t+1} = \theta_t - \eta \nabla F(\theta_t)$$

(or some variant thereof, e.g. Adam)

Functional Optimization

$$\operatorname*{argmin}_{f\,:\,\mathcal{X}\to\mathcal{Y}} \mathcal{R}(f)$$

$$\operatorname*{argmin}_{\theta\in\mathbb{R}^d} F(\theta), \qquad F(\theta) = \mathcal{R}(f_\theta)$$

Loss & gradient operator:

$$F : \mathbb{R}^d \to \mathbb{R}, \qquad \nabla F : \mathbb{R}^d \to \mathbb{R}^d$$

Optimize with gradient descent:

$$\theta_{t+1} = \theta_t - \eta \nabla F(\theta_t)$$

(or some variant thereof, e.g. Adam)

$$\operatorname*{argmin}_{h\in\mathcal{H}} \mathcal{R}(h), \qquad \mathcal{H} \text{ Hilbert}$$

Functional Optimization

$$\operatorname*{argmin}_{f\,:\,\mathcal{X}\to\mathcal{Y}} \mathcal{R}(f)$$

$$\operatorname*{argmin}_{\theta\in\mathbb{R}^d} F(\theta), \qquad F(\theta) = \mathcal{R}(f_\theta)$$

Loss & gradient operator:

$$F : \mathbb{R}^d \to \mathbb{R}, \qquad \nabla F : \mathbb{R}^d \to \mathbb{R}^d$$

Optimize with gradient descent:

$$\theta_{t+1} = \theta_t - \eta \nabla F(\theta_t)$$

(or some variant thereof, e.g. Adam)

$$\operatorname*{argmin}_{h\in\mathcal{H}} \mathcal{R}(h), \qquad \mathcal{H} \text{ Hilbert}$$

Loss & gradient operator:

$$\mathcal{R} : \mathcal{H} \to \mathbb{R}, \qquad \nabla \mathcal{R} : \mathcal{H} \to \mathcal{H}$$

Functional Optimization

$$\operatorname*{argmin}_{f\,:\,\mathcal{X}\to\mathcal{Y}} \mathcal{R}(f)$$

$$\operatorname*{argmin}_{\theta\in\mathbb{R}^d} F(\theta), \qquad F(\theta) = \mathcal{R}(f_\theta)$$

Loss & gradient operator:

$$F : \mathbb{R}^d \to \mathbb{R}, \qquad \nabla F : \mathbb{R}^d \to \mathbb{R}^d$$

Optimize with gradient descent:

$$\theta_{t+1} = \theta_t - \eta \nabla F(\theta_t)$$

(or some variant thereof, e.g. Adam)

$$\operatorname*{argmin}_{h\in\mathcal{H}} \mathcal{R}(h), \qquad \mathcal{H} \text{ Hilbert}$$

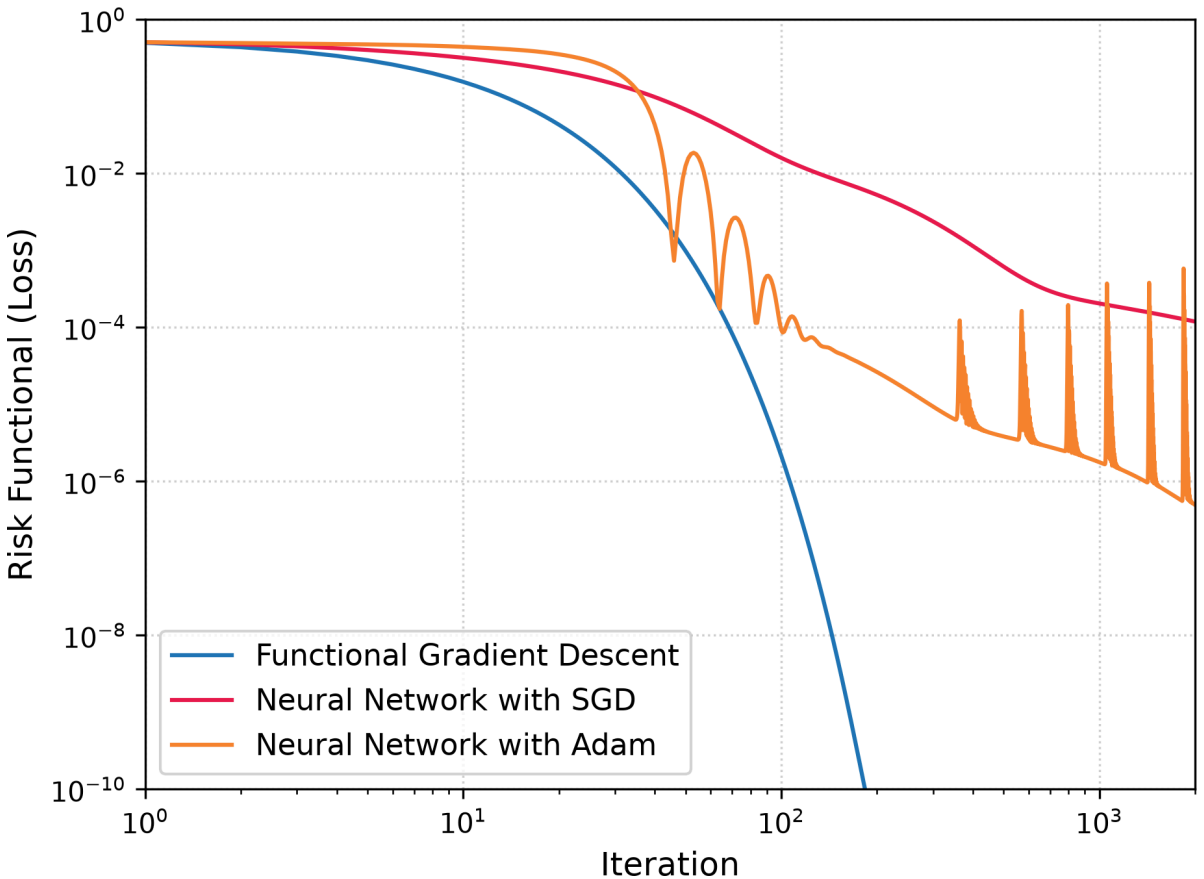
Loss & gradient operator:

$$\mathcal{R} : \mathcal{H} \to \mathbb{R}, \qquad \nabla \mathcal{R} : \mathcal{H} \to \mathcal{H}$$

Optimize with *functional* gradient descent:

$$h_{t+1} = h_t - \eta \nabla \mathcal{R}(h_t)$$

Functional Optimization



Functional Gradients are Hard

Problem: it's really hard to compute functional gradients.

Functional Gradients are Hard

Problem: it's really hard to compute functional gradients.

Running example: Solving a PDE

$$\begin{cases} L[u](x) = 0 & x \in \Omega \\ u(x) = f(x) & x \in \partial\Omega \end{cases}$$

where L is a differential operator; consider the risk functional

$$\mathcal{R}(h) : \mathbb{H}^s(\Omega) \to L^2(\Omega)$$
$$h \mapsto \|L[h]\|^2_{L^2(\Omega)} + \|h - f\|^2_{L^2(\partial\Omega)}$$

Functional Gradients are Hard

Full gradients are *hard*: $\nabla \mathcal{R}(h)$ is the unique $g \in \mathbb{H}^s(\Omega)$ such that

$$\sum_{j=0}^{s} \int_\Omega \mathrm{D}^{(j)} g(x) \cdot \mathrm{D}^{(j)} v(x)\, \mathrm{d}x = \int_\Omega L[h](x) L[v](x)\, \mathrm{d}x + \int_{\partial\Omega} (h(x) - f(x)) v(x)\, \sigma(\mathrm{d}x)$$

$$\text{for all } v \in \mathbb{H}^s(\Omega).$$

Functional Gradients are Hard

Full gradients are *hard*: $\nabla\mathcal{R}(h)$ is the unique $g \in \mathbb{H}^s(\Omega)$ such that

$$\sum_{j=0}^{s} \int_\Omega \mathrm{D}^{(j)} g(x) \cdot \mathrm{D}^{(j)} v(x)\,\mathrm{d}x = \int_\Omega L[h](x) L[v](x)\,\mathrm{d}x + \int_{\partial\Omega} (h(x) - f(x)) v(x)\,\sigma(\mathrm{d}x)$$

$$\text{for all } v \in \mathbb{H}^s(\Omega).$$

But directional derivatives are easy:

$$\begin{aligned} \mathrm{D}\mathcal{R}(h; v) &:= \lim_{\delta \to 0} \frac{\mathcal{R}(h + \delta v) - \mathcal{R}(h)}{\delta} \\ &= \left[\frac{\mathrm{d}}{\mathrm{d}\delta} \mathcal{R}(h + \delta v) \right]_{\delta = 0} \end{aligned}$$

Grad. Descent with Directional Derivatives

Q: Can we use just directional derivatives for optimization?

Grad. Descent with Directional Derivatives

Q: Can we use just directional derivatives for optimization?

A: In finite dimensions, yes: Random ‘Gradient-Free’ Optimization

Grad. Descent with Directional Derivatives

Q: Can we use just directional derivatives for optimization?

A: In finite dimensions, yes: Random 'Gradient-Free' Optimization

To estimate $\nabla \mathcal{R}(h)$, sample $v_1, ..., v_M \sim \mathbb{P}_v$, and compute: $h, v_1, ..., v_M \in \mathbb{R}^d$

$$\hat{g} = \frac{1}{M} \sum_{m=1}^{M} \mathrm{D}\mathcal{R}(h; v_m)\, v_m$$

Grad. Descent with Directional Derivatives

Q: Can we use just directional derivatives for optimization?

A: In finite dimensions, yes: Random 'Gradient-Free' Optimization

To estimate $\nabla \mathcal{R}(h)$, sample $v_1, ..., v_M \sim \mathbb{P}_v$, and compute: $h, v_1, ..., v_M \in \mathbb{R}^d$

$$\hat{g} = \frac{1}{M} \sum_{m=1}^{M} \mathrm{D}\mathcal{R}(h; v_m)\, v_m$$

- $\mathbb{E}[v] = 0$ and $\mathrm{Cov}[v] = I \quad \Longrightarrow \quad \mathbb{E}[\hat{g}] = \nabla L(h).$ (e.g. $\mathbb{P}_v = \mathcal{N}(0, I)$)

Grad. Descent with Directional Derivatives

Q: Can we use just directional derivatives for optimization?

A: In finite dimensions, yes: Random 'Gradient-Free' Optimization

To estimate $\nabla \mathcal{R}(h)$, sample $v_1, ..., v_M \sim \mathbb{P}_v$, and compute: $h, v_1, ..., v_M \in \mathbb{R}^d$

$$\hat{g} = \frac{1}{M} \sum_{m=1}^{M} \mathrm{D}\mathcal{R}(h; v_m)\, v_m$$

- $\mathbb{E}[v] = 0$ and $\mathrm{Cov}[v] = I \quad \Longrightarrow \quad \mathbb{E}[\hat{g}] = \nabla L(h).$ (e.g. $\mathbb{P}_v = \mathcal{N}(0, I)$)

However, this breaks in infinite dimensions:

- Variance blows up as $d \to +\infty$
- There is no isotropic Gaussian $\mathcal{N}(0, I)$, nor analogous distributions
- Even if we change the distribution, $\mathbb{E}[\|v\|^2] < +\infty \Longrightarrow \mathrm{Cov}[v] \neq I$.

Infinite Dim. Grad.-Free Descent

Ingredients:

- A prebasis for the Hilbert space: $\mathcal{B} = \{b_1, b_2, ...\}$ such that

$$\mathcal{B} \text{ is linearly independent} \qquad \text{and} \qquad \text{span}(\mathcal{B}) \text{ is dense in } \mathcal{H}$$

- A distribution $\mathbb{P}_K$ over the positive naturals;

Infinite Dim. Grad.-Free Descent

Ingredients:

- A prebasis for the Hilbert space: $\mathcal{B} = \{b_1, b_2, \ldots\}$ such that

$$\mathcal{B} \text{ is linearly independent} \qquad \text{and} \qquad \operatorname{span}(\mathcal{B}) \text{ is dense in } \mathcal{H}$$

- A distribution $\mathbb{P}_K$ over the positive naturals;

The procedure: To estimate $\nabla \mathcal{R}(h)$:

- Sample a dimension $K \sim \mathbb{P}_K$, and choose a batch size $M_K \in \mathbb{N}$ (e.g. $\lceil K/2 \rceil$)

Infinite Dim. Grad.-Free Descent

Ingredients:

- A prebasis for the Hilbert space: $\mathcal{B} = \{b_1, b_2, ...\}$ such that

$$\mathcal{B} \text{ is linearly independent} \qquad \text{and} \qquad \text{span}(\mathcal{B}) \text{ is dense in } \mathcal{H}$$

- A distribution $\mathbb{P}_K$ over the positive naturals;

The procedure: To estimate $\nabla \mathcal{R}(h)$:

- Sample a dimension $K \sim \mathbb{P}_K$, and choose a batch size $M_K \in \mathbb{N}$ (e.g. $\lceil K/2 \rceil$)
- Sample weights $\boldsymbol{w}^{(1)}, ..., \boldsymbol{w}^{(M_K)} \sim \mathcal{N}(0, \Sigma_K)$ in $\mathbb{R}^K$, with a tailored Σ_K

Infinite Dim. Grad.-Free Descent

Ingredients:

- A prebasis for the Hilbert space: $\mathcal{B} = \{b_1, b_2, \ldots\}$ such that

$$\mathcal{B} \text{ is linearly independent} \qquad \text{and} \qquad \operatorname{span}(\mathcal{B}) \text{ is dense in } \mathcal{H}$$

- A distribution $\mathbb{P}_K$ over the positive naturals;

The procedure: To estimate $\nabla \mathcal{R}(h)$:

- Sample a dimension $K \sim \mathbb{P}_K$, and choose a batch size $M_K \in \mathbb{N}$ (e.g. $\lceil K/2 \rceil$)
- Sample weights $\boldsymbol{w}^{(1)}, \ldots, \boldsymbol{w}^{(M_K)} \sim \mathcal{N}(0, \Sigma_K)$ in $\mathbb{R}^K$, with a tailored Σ_K
- Compute

$$\hat{g} = \frac{\lambda_K}{M_K} \sum_{m=1}^{M_K} \mathrm{D}\mathcal{R}(h; v_m)\, v_m, \qquad \text{where} \qquad v_i := \sum_{j=1}^{K} \boldsymbol{w}_j^{(i)} b_j$$

Infinite Dim. Grad.-Free Descent

Ingredients:

- A prebasis for the Hilbert space: $\mathcal{B} = \{b_1, b_2, \ldots\}$ such that

$$\mathcal{B} \text{ is linearly independent} \qquad \text{and} \qquad \operatorname{span}(\mathcal{B}) \text{ is dense in } \mathcal{H}$$

- A distribution $\mathbb{P}_K$ over the positive naturals;

The procedure: To estimate $\nabla \mathcal{R}(h)$:

- Sample a dimension $K \sim \mathbb{P}_K$, and choose a batch size $M_K \in \mathbb{N}$ (e.g. $\lceil K/2 \rceil$)
- Sample weights $\boldsymbol{w}^{(1)}, \ldots, \boldsymbol{w}^{(M_K)} \sim \mathcal{N}(0, \Sigma_K)$ in $\mathbb{R}^K$, with a tailored Σ_K
- Compute

$$\hat{g} = \frac{\lambda_K}{M_K} \sum_{m=1}^{M_K} \mathrm{D}\mathcal{R}(h; v_m)\, v_m, \qquad \text{where} \qquad v_i := \sum_{j=1}^{K} \boldsymbol{w}_j^{(i)} b_j$$

Infinite Dim. Grad.-Free Descent

Proposition. $\hat{g}$ is an unbiased estimator of a preconditioned gradient:

$$\mathbb{E}[\hat{g}] = C\,\nabla\mathcal{R}(h),$$

where $C : \mathcal{H} \to \mathcal{H}$ is a positive-definite diagonal operator determined by $(\lambda_1, \lambda_2, \ldots)$.

Infinite Dim. Grad.-Free Descent

Proposition. $\hat{g}$ is an unbiased estimator of a preconditioned gradient:

$$\mathbb{E}[\hat{g}] = C\,\nabla\mathcal{R}(h),$$

where $C : \mathcal{H} \to \mathcal{H}$ is a positive-definite diagonal operator determined by $(\lambda_1, \lambda_2, \ldots)$.

If we take $\lambda_k \equiv 1$, then we get $C = I$. But the **variance explodes**:

Proposition. If $\lambda_k \equiv 1$, then

$$\mathbb{E}\big[\|\hat{g}\|^2\big] \geq \sum_{i=1}^{\infty} \frac{\langle \nabla\mathcal{R}(h), e_i\rangle^2}{t_i}.$$

Infinite Dim. Grad.-Free Descent

Proposition. $\hat{g}$ is an unbiased estimator of a preconditioned gradient:

$$\mathbb{E}[\hat{g}] = C\,\nabla\mathcal{R}(h),$$

where $C : \mathcal{H} \to \mathcal{H}$ is a positive-definite diagonal operator determined by $(\lambda_1, \lambda_2, \ldots)$.

So we take $\lambda_k = \mathbb{P}[K \geq k]$:

Proposition. Take $\lambda_k = \mathbb{P}[K \geq k]$ and $M_k = \lceil k/c \rceil$, for some $c > 0$. We then have

$$\mathbb{E}[\|\hat{g}\|_C^2] \leq 2(1 + 2c)\,\|\nabla\mathcal{R}(h)\|^2,$$

where $\|\hat{g}\|_C := \left\|C^{-\frac{1}{2}}\hat{g}\right\|$.

Convergence

Theorem. If the risk is convex, has L-Lipschitz gradients and the minimizer $h^\star$ is reasonably regular, then for a sufficiently small learning rate α,

$$\mathbb{E}\left[\mathcal{R}\left(\hat{h}\right)\right] - \mathcal{R}(h^\star) \leq \|h^\star - h_1\|_C^2 \cdot \left(\frac{1}{2N\alpha} + \alpha(1+2c)L^2\right),$$

where $\hat{h}$ is the averaged iterate $\frac{1}{N}(h_1 + \cdots + h_N)$.

Convergence

Theorem. If the risk is convex, has L-Lipschitz gradients and the minimizer $h^\star$ is reasonably regular, then for a sufficiently small learning rate α,

$$\mathbb{E}\left[\mathcal{R}\left(\hat{h}\right)\right] - \mathcal{R}(h^\star) \leq \|h^\star - h_1\|_C^2 \cdot \left(\frac{1}{2N\alpha} + \alpha(1+2c)L^2\right),$$

where $\hat{h}$ is the averaged iterate $\frac{1}{N}(h_1 + \cdots + h_N)$.

Corollary. Choosing $\alpha = O\left(1/\sqrt{N}\right)$, we get

$$\mathbb{E}\left[\mathcal{R}\left(\hat{h}\right)\right] - \mathcal{R}(h^\star) \leq O\left(\frac{1}{\sqrt{N}}\right).$$

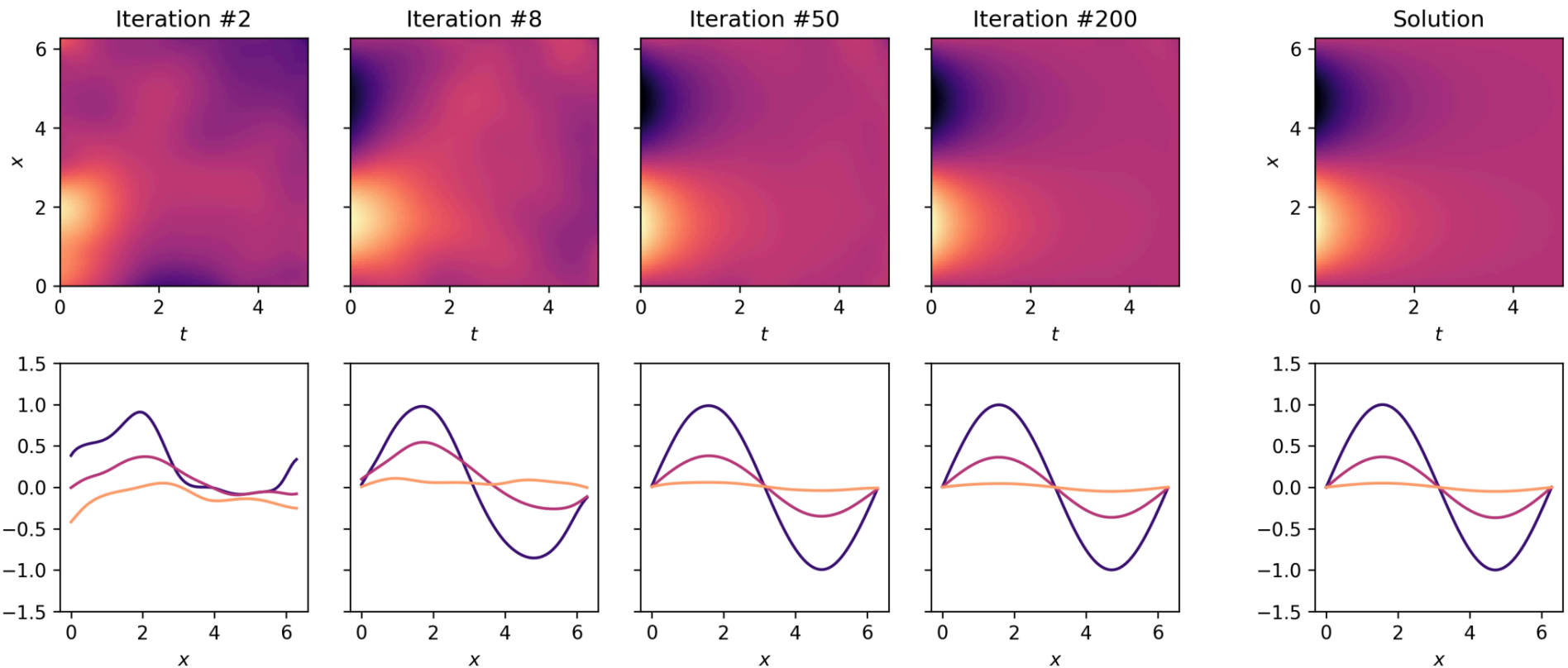
Example 1: Heat Equation

A simple heat equation:

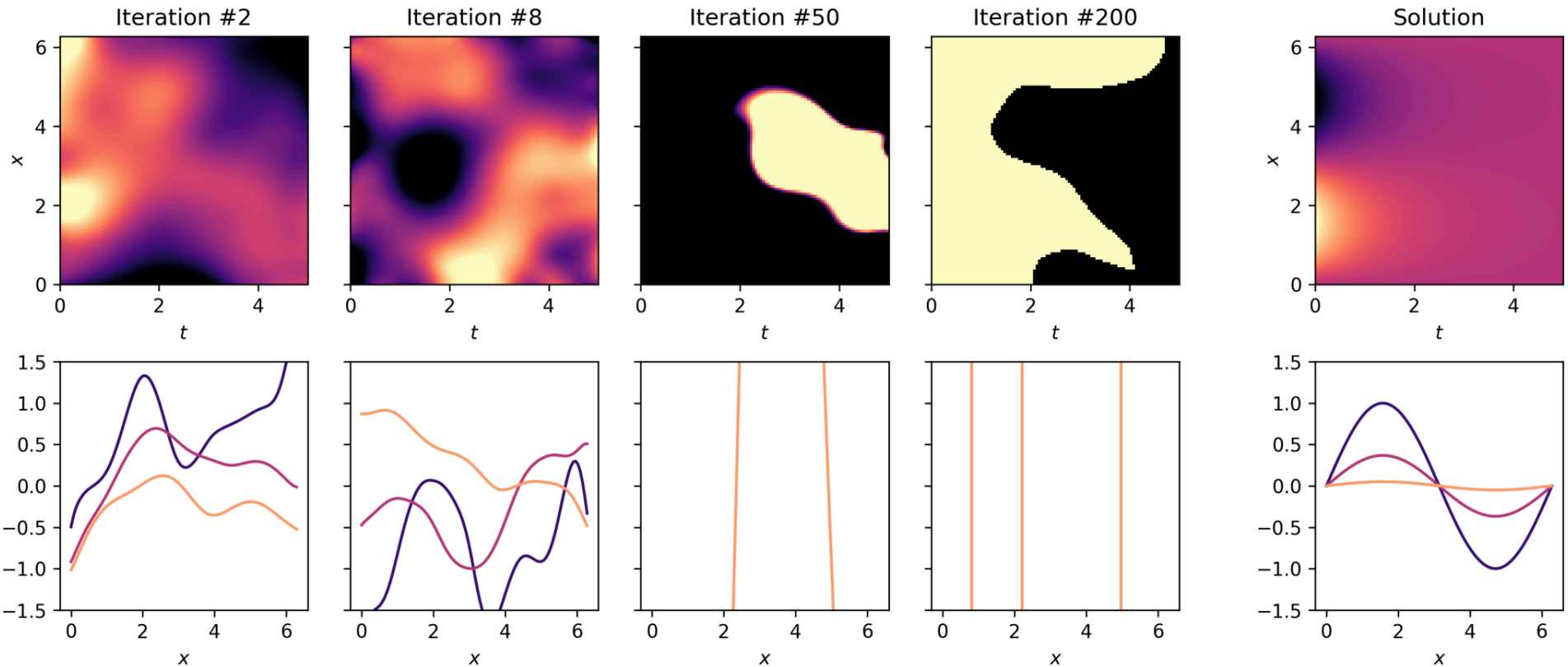
$$\begin{cases} \partial_t u(t,x) = \Delta u(t,x) & t \in (0,T), x \in (0,2\pi) \\ u(0,x) = \sin x & x \in (0,2\pi) \\ u(t,0) = u(t,2\pi) = 0 & t \in (0,T) \end{cases}$$

Unique solution given by $u(t,x) = e^{-t} \sin x$

Example 1: Heat Equation

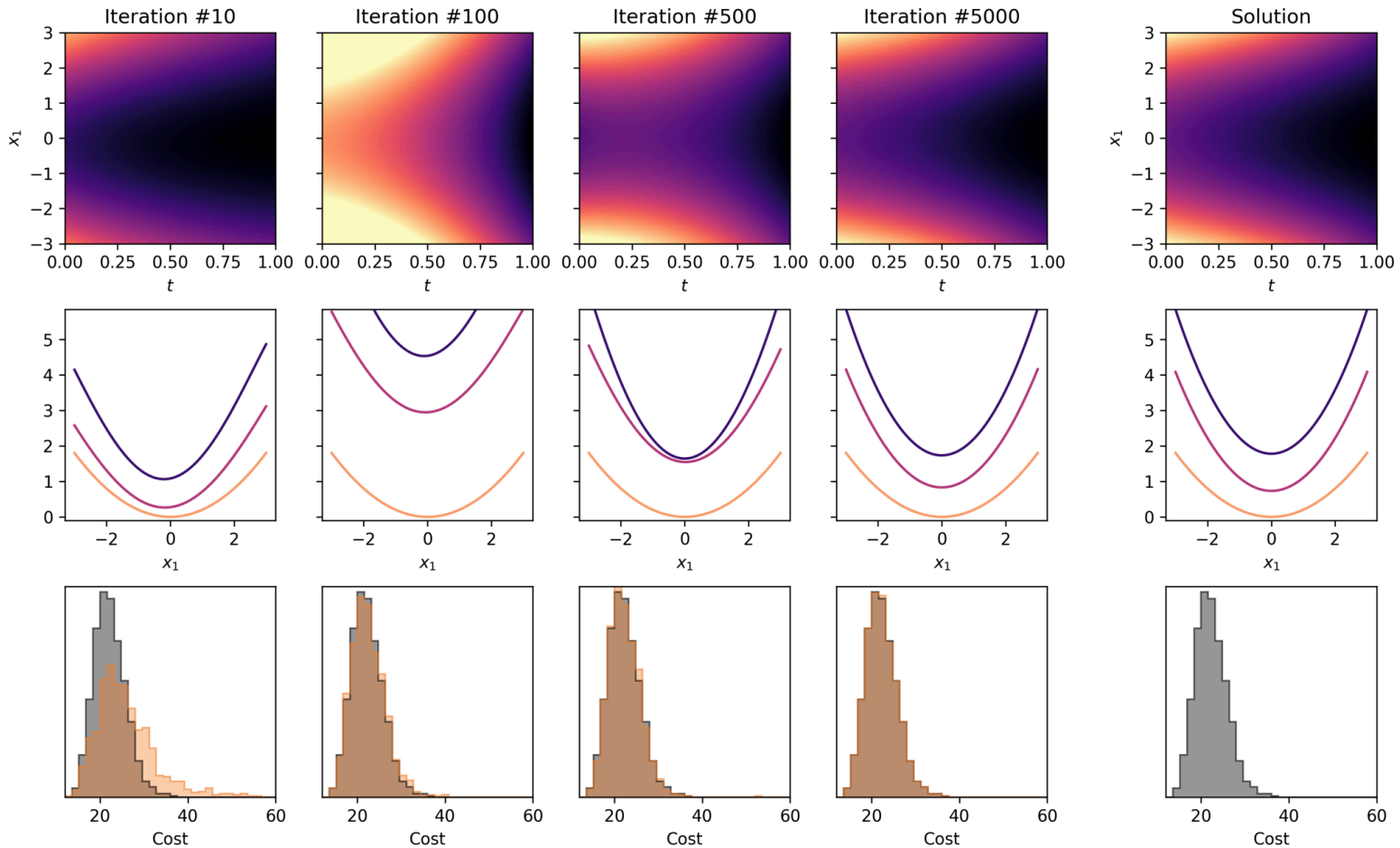


Example 1: Heat Equation



without preconditioning ($\lambda_k \equiv 1$)

Example 2: HJB Equation in 5+1 dims.



Conclusion

- Functional gradient descent is effective, but computing functional gradients is hard
  - Our method allows for estimation of functional gradients with autodiff
- Also in the paper: more general statements, some impossibility results, approximation error of $\mathbb{P}_K$, prebasis for Sobolev spaces, more examples, *etc.*
- **Extensions:** accelerated variants? proximal variants? variance reduction?

Caio Peixoto, Daniel Csillag, Bernardo Costa and Yuri Saporito. *Random Gradient-Free Optimization in Infinite Dimensional Spaces*

See also: Daniel Csillag, Rodrigo Schuller, Pedro Dall'Antonia, Leonidas Guibas, Luiz Velho, Tiago Novello. *Functional Gradient Descent with Adaptive Representations*

Conclusion

- Functional gradient descent is effective, but computing functional gradients is hard
  - Our method allows for estimation of functional gradients with autodiff
- Also in the paper: more general statements, some impossibility results, approximation error of $\mathbb{P}_K$, prebasis for Sobolev spaces, more examples, *etc.*
- **Extensions:** accelerated variants? proximal variants? variance reduction?

Caio Peixoto, Daniel Csillag, Bernardo Costa and Yuri Saporito. *Random Gradient-Free Optimization in Infinite Dimensional Spaces*

See also: Daniel Csillag, Rodrigo Schuller, Pedro Dall'Antonia, Leonidas Guibas, Luiz Velho, Tiago Novello. *Functional Gradient Descent with Adaptive Representations*

supplementary

Infinite Dim. Grad.-Free Descent

Key ingredients:

1. A prebasis for the Hilbert space: $\mathcal{B} = \{b_1, b_2, ...\}$ such that
$$\mathcal{B} \text{ is linearly independent} \qquad \text{and} \qquad \operatorname{span}(\mathcal{B}) \text{ is dense in } \mathcal{H}.$$
2. Tail probabilities $t_i := \mathbb{P}[K \geq i]$.

To sample $v|K$:

1. Compute the Gram matrix $[\boldsymbol{A}_K]_{i,j} := \langle b_i, b_j \rangle$
2. Find its Cholesky decomposition $\boldsymbol{R}_K^\top \boldsymbol{R}_K = \boldsymbol{A}_K$.
   (Equivalent to finding the generalized QR decomposition $Q_K \boldsymbol{R}_K$ of $[b_1 \ \cdots \ b_K]$.)
3. Sample weights $\boldsymbol{w} \sim \mathcal{N}\left(0, \boldsymbol{R}_K^{-1} \boldsymbol{T}_K^{-1} \boldsymbol{R}_K^{-\top}\right)$, where $\boldsymbol{T}_K := \operatorname{diag}(t_1, ..., t_K)$, and take
$$v = \sum_{i=1}^{K} \boldsymbol{w}_i b_i.$$