# An Interpolation Perspective on Neural Network Generalization

Jean-Michel Morel
Joint work with Jin GUO and Roy HE

Lingnan University

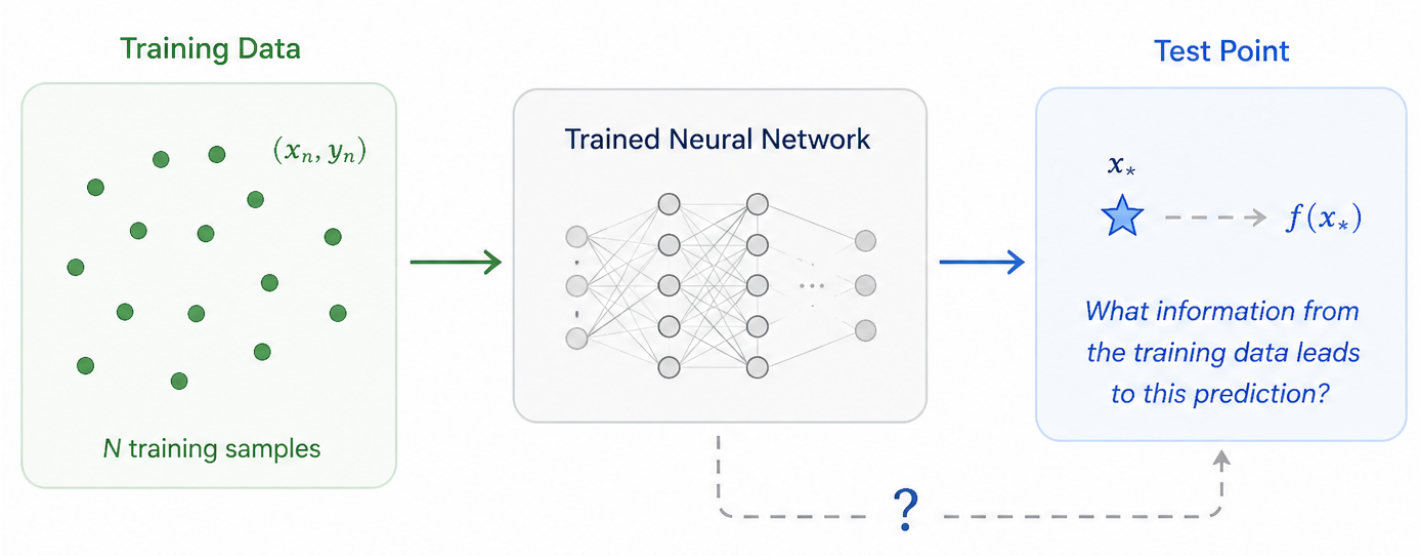
FAU, Erlangen

June 21, 2026

# Content

# Motivation

How does a trained neural network generalize?




We consider any parametric machine trained by gradient descent on a dataset of examples of (input, output) pairs. Gradient descent is applied to a training loss expressing how well a network fits the observed samples. The question of generalization is how a trained network behaves at an unseen input, namely how it relates the input to the training inputs.
The Domingos formula gives an explicit answer to this question.

# Domingos Formula: A Path-Kernel View

Consider the general training setup:

- Dataset $\{(x_n, y_n^*)\}_{n=1}^N$, neural network $f(x, \Theta) : \mathbb{R}^p \times \mathbb{R}^d \to \mathbb{R}^m$.
- Loss: $L(\Theta) = \frac{1}{N} \sum_{n=1}^N \ell(f(x_n, \Theta), y_n^*)$.
- Training dynamics: $\Theta_k = \Theta_{k-1} - \eta \nabla_\Theta L(\Theta_{k-1})$.
- The associated continuous gradient flow, or learning dynamics is defined by

$$\dot{\Theta}_t = -\nabla L(\Theta_t)$$

We define the *path kernel* associated to the learning dynamics by

$$K_t(x, x') := \langle \nabla_\Theta f(x, \Theta_t), \nabla_\Theta f(x', \Theta_t) \rangle.$$

# Domingos Formula: A Path-Kernel View

Consider the general training setup:

- Dataset $\{(x_n, y_n^*)\}_{n=1}^N$, neural network $f(x, \Theta) : \mathbb{R}^p \times \mathbb{R}^d \to \mathbb{R}^m$.
- Loss: $L(\Theta) = \frac{1}{N} \sum_{n=1}^N \ell(f(x_n, \Theta), y_n^*)$.
- Training dynamics: $\Theta_k = \Theta_{k-1} - \eta \nabla_\Theta L(\Theta_{k-1})$.

When the learning rate $\eta \to 0$ and $\eta k \to T$, the Domingos formula[1] is

$$\lim_{\eta \to 0} f(x, \Theta_T) = f(x, \Theta_0) - \frac{1}{N} \sum_{n=1}^N \int_0^T \underbrace{\langle \nabla_\Theta f(x, \Theta_t), \nabla_\Theta f(x_n, \Theta_t) \rangle}_{\text{interaction weight; path kernel}} \underbrace{\frac{\partial \ell}{\partial y_n}(x_n, \Theta_t)}_{\text{training correction}} \, \mathrm{d}t.$$

The path kernel measures how strongly the test point interacts with the training sample in the tangent feature space at time $t$.

[1] Pedro Domingos. Every model learned by gradient descent is approximately a kernel machine. arXiv preprint arXiv:2012.00152, 2020.

# Domingos Formula: an interpolation formula

For a quadratic loss, the Domingos formula becomes

$$f(x,\Theta_T) = f(x,\Theta_0) - \frac{1}{N}\sum_{n=1}^{N}\int_0^T \underbrace{\langle \nabla_\Theta f(x,\Theta_t), \nabla_\Theta f(x_n,\Theta_t)\rangle}_{\text{interaction weight//path kernel}} \underbrace{(f(x_n,\Theta_t) - y_n^*)}_{\text{training residual}} \,\mathrm{d}t.$$

This formula makes it apparent that the output of the learning machine for the input $x$ is a weighted average along the path of the residuals of all training samples, weighted by the path kernel. The path kernel appears like a weight that measures the interaction between the input $x$ and the other samples $x_n$

# Domingos Theorem

**Theorem (Informal)** Let $f(\cdot, \Theta)$ be a differentiable model trained by gradient descent $\dot{\Theta}_t = -\nabla L(\Theta_t)$ on the data $\{(x_n, y_n^*)\}_{n=1}^N$, using the empirical loss $L(\Theta) = \frac{1}{N}\sum_{n=1}^N \ell(f(x_n, \Theta), y_n^*)$. Then for any input $x \in \mathbb{R}^p$, the model output at time $T$ satisfies

$$f(x, \Theta_T) = f(x, \Theta_0) - \frac{1}{N}\sum_{n=1}^N \int_0^T K_t(x, x_n) \frac{\partial \ell}{\partial f}\left(f(x_n, \Theta_t), y_n^*\right) dt, \tag{1}$$

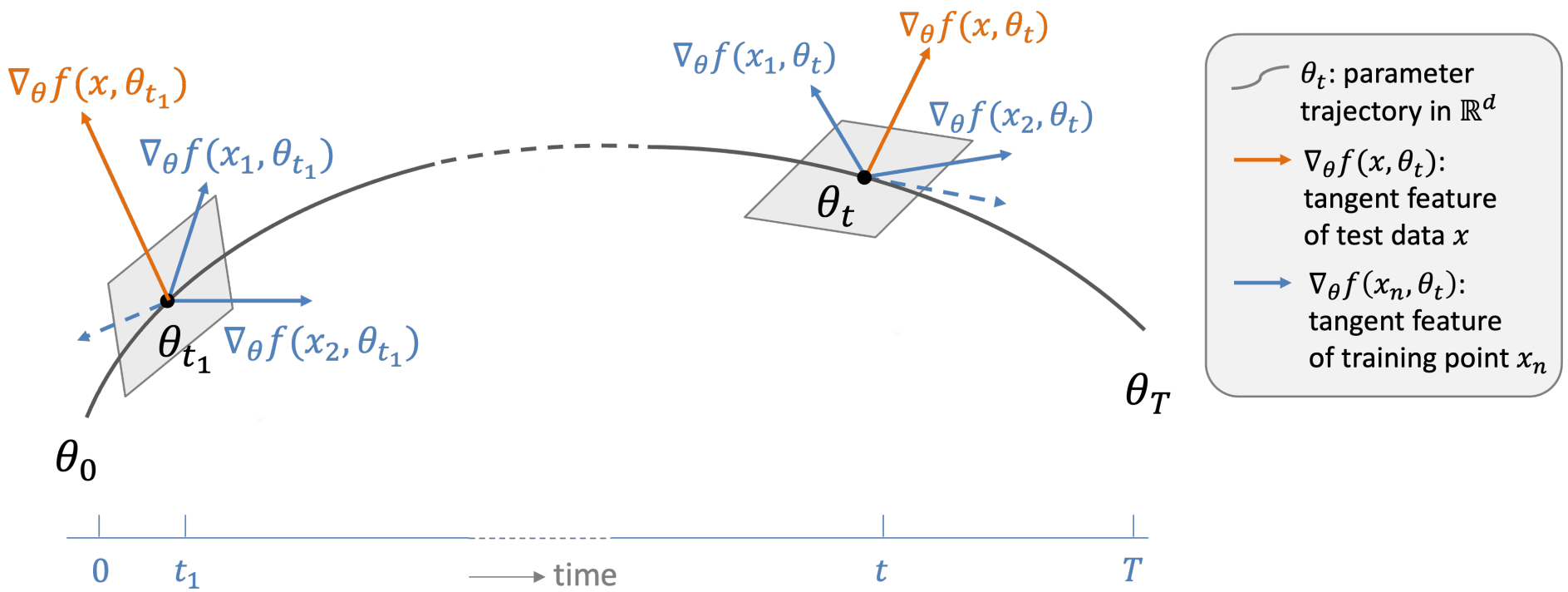
**Proof.** Let $\Theta_t = (\Theta_t^1, \cdots, \Theta_t^d) \in \mathbb{R}^d$ denote the parameter of the neural network $f(\cdot, \Theta)$. By the chain rule and the gradient flow $\dot{\Theta}_t = -\nabla L(\Theta_t)$, we have

$$\begin{aligned}
\frac{\mathrm{d}f}{\mathrm{d}t}(x, \Theta_t) &= -\sum_{i=1}^d \frac{\partial f}{\partial \Theta_t^i}(x, \Theta_t) \frac{\partial L(\Theta_t)}{\partial \Theta_t^i} \\
&= -\sum_{i=1}^d \frac{\partial f}{\partial \Theta_t^i}(x, \Theta_t) \left( \frac{1}{N}\sum_{n=1}^N \frac{\partial f}{\partial \Theta_t^i}(x_n, \Theta_t) \frac{\partial \ell}{\partial f}(f(x_n, \Theta_t), y_n^*) \right) \\
&= -\frac{1}{N}\sum_{n=1}^N \sum_{i=1}^d \frac{\partial f}{\partial \Theta_t^i}(x, \Theta_t) \frac{\partial f}{\partial \Theta_t^i}(x_n, \Theta_t) \frac{\partial \ell}{\partial f}(f(x_n, \Theta_t), y_n^*) \\
&= -\frac{1}{N}\sum_{n=1}^N \langle \nabla_\Theta f(x, \Theta_t), \nabla_\Theta f(x_n, \Theta_t) \rangle \frac{\partial \ell}{\partial f}(f(x_n, \Theta_t), y_n^*).
\end{aligned}$$

Integrating both sides from 0 to $T$ gives (1).

# Domingos Theorem

$$f(x, \Theta_T) = f(x, \Theta_0) - \frac{1}{N} \sum_{n=1}^{N} \int_0^T \underbrace{\langle \nabla_\Theta f(x, \Theta_t), \nabla_\Theta f(x_n, \Theta_t) \rangle}_{\text{interaction weight}} \underbrace{(f(x_n, \Theta_t) - y_n^*)}_{\text{training residual}} \, \mathrm{d}t.$$



# Is Domingos formula general?

- What if training is made by stochastic gradient descent: Real training is done by stochastic optimizers using random mini batches.
- Is there a path kernel representation for stochastic optimizers?
- How does the Domingos formula depend on the kind of stochastic optimization (SGDM, RMSprop, Adam)?
- What does this imply for generalization?

# Continuous Time Approximation of the discrete SGD

The simplest stochastic gradient descent (SGD) is defined by

$$\Theta_{k+1} = \Theta_k - \eta \nabla_\Theta L_{\mathcal{B}_k}(\Theta_k),$$

where $\mathcal{B}_k$ is a random uniform mini-batch with fixed cardinality and

$$L_{\mathcal{B}_k}(\Theta_k) = \frac{1}{|\mathcal{B}_k|} \sum_{n \in \mathcal{B}_k} \ell(f(x_n, \Theta_k), y_n^*).$$

To generalize Domingos formula, we need a continuous model and an asymptotic formulation of the learning dynamics as a stochastic differential equation.
(Several propositions for a continuous SDE [Mandt, 2015; Jastrzebski, 2018, Zhu, 2019])

# Stochastic Domingos Theorem

**Lemma** (First-order weak approximation of SGD[2]) Let $T > 0$, $\eta \in (0, 1 \wedge T)$, and set $K = \lfloor T/\eta \rfloor$. Let $\{\Theta_k\}_{k \geq 0}$ be the SGD iterates. Define $\{Z_t : t \in [0, T]\}$ as the solution of the SDE

$$\mathrm{d}Z_t = -\nabla L(Z_t)\,\mathrm{d}t + \sqrt{\eta}\,\Sigma^{1/2}_{|\mathcal{B}|}(Z_t)\,\mathrm{d}W_t, \qquad Z_0 = \Theta_0, \tag{2}$$

where

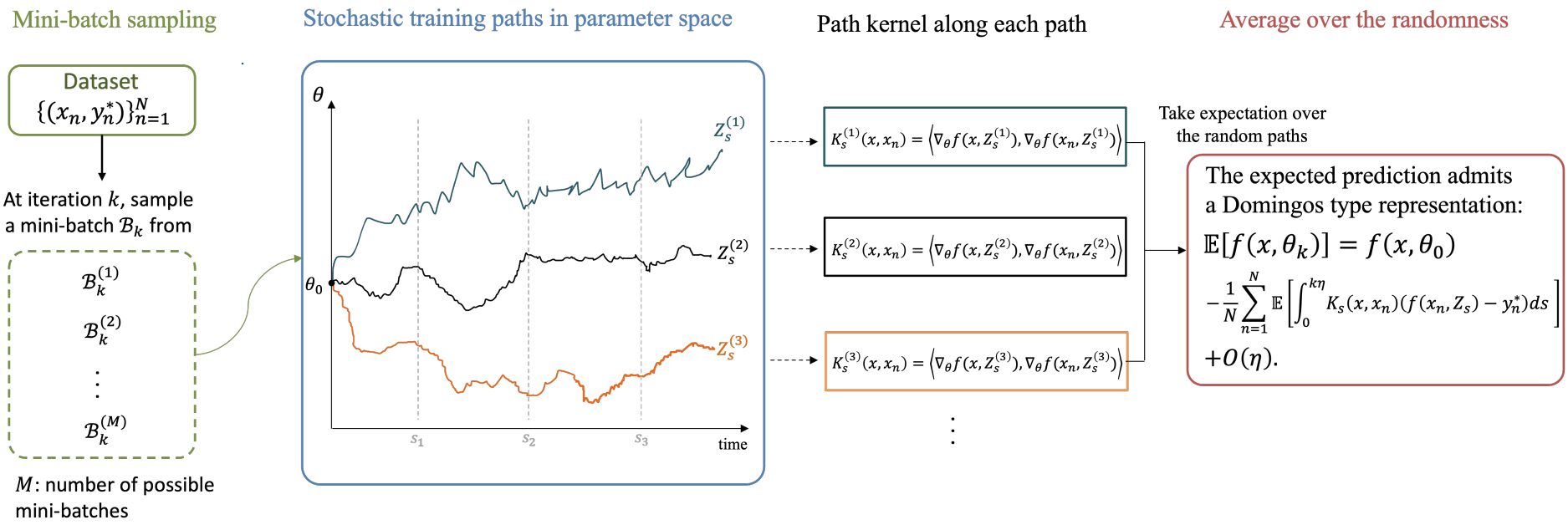
$$\Sigma_{|\mathcal{B}|}(Z) = \mathbb{E}_{\mathcal{B}}\big[\big(\nabla L_{\mathcal{B}}(Z) - \nabla L(Z)\big)\big(\nabla L_{\mathcal{B}}(Z) - \nabla L(Z)\big)^{\top}\big].$$

Then $\{Z_t\}$ is a first-order weak approximation of the SGD iterates; for any smooth test function $g$,

$$\max_{k=0,\ldots,K} \big|\mathbb{E}[g(\Theta_k)] - \mathbb{E}[g(Z_{k\eta})]\big| = O(\eta). \tag{3}$$

[2] Qianxiao Li, Cheng Tai, and E Weinan. Stochastic modified equations and dynamics of stochastic gradient algorithms I: Mathematical foundations. Journal of Machine Learning Research, 20(40):1–47, 2019.

# Stochastic Domingos Theorem




**Theorem** (Informal) Consider a learning model $y = f(x, \Theta)$. Assume that $\Theta$ is learned from a dataset $\{(x_n, y_n^*)\}_{n=1}^N$ by SGD with learning rate $\eta$. Then

$$\mathbb{E}\big[f(x, \Theta_k)\big] = f(x, \Theta_0) - \frac{1}{N}\sum_{n=1}^{N} \mathbb{E}\left[\int_0^{k\eta} \underbrace{K_s(x, x_n)}_{\text{interaction weight}} \underbrace{\frac{\partial \ell}{\partial f}\big(f(x_n, Z_s), y_n^*\big)}_{\text{training correction}} \, \mathrm{d}s\right] + O(\eta), \qquad (4)$$

where $K_s(x, x') := \big\langle \nabla_\Theta f(x, Z_s),\, \nabla_\Theta f(x', Z_s) \big\rangle$ is the stochastic path kernel.

# Stochastic Domingos' Theorem (SGDM)

Stochastic Gradient Descent with Momentum

**Theorem** (Informal) Consider a learning model $y = f(x, \Theta)$. Assume that $\Theta$ is learned from the dataset $\{(x_n, y_n^*)\}_{n=1}^N$ by SGDM with learning rate $\eta$ and momentum weight $\beta \in [0, 1)$. Let $\mu = (1 - \beta)/\eta$. Then

$$\mathbb{E}[f(x, \Theta_k)] = f(x, \Theta_0) - \frac{1}{N} \sum_{n=1}^{N} \mathbb{E}\left[ \int_0^{k\eta} \int_0^t \underbrace{K_{t,s}(x, x_n)}_{\substack{\text{interaction with} \\ \text{memory decay}}} \underbrace{\frac{\partial \ell}{\partial f}\left(f(x_n, Z_s), y_n^*\right)}_{\text{training correction}} \mathrm{d}s\, \mathrm{d}t \right] + O(\eta), \tag{5}$$

where the path kernel has become
$K_{t,s}(x, x') := \left\langle \nabla_\Theta f(x, Z_t),\ e^{-(t-s)\mu} \nabla_\Theta f(x', Z_s) \right\rangle$.

When $\mu$ is large (weak momentum), the memory kernel collapses and the result turns to the SGD case.

# Stochastic Domingos' Theorem (RMSprop)

**Theorem** (Informal) Consider a learning model $y = f(x, \Theta)$. Assume that $\Theta$ is learned from the dataset $\{(x_n, y_n^*)\}_{n=1}^N$ by RMSprop (Root Mean Square Propagation) with learning rate $\eta$, second-moment decay rate $\beta \in [0, 1)$, and smoothing constant $\varepsilon > 0$. Let $\mu = (1 - \beta)/\eta$ and suppose that $\mu = O(1)$ as $\eta \to 0$. Then

$$\mathbb{E}[f(x, \Theta_k)] = f(x, \Theta_0) - \frac{1}{N} \sum_{n=1}^{N} \mathbb{E}\Bigg[ \int_0^{k\eta} \underbrace{K_t^{P_t}(x, x_n)}_{\substack{\text{preconditioned} \\ \text{interaction weight}}} \underbrace{\frac{\partial \ell}{\partial f}\big(f(x_n, Z_t), y_n^*\big)}_{\text{training correction}} \, \mathrm{d}t \Bigg]$$

$$+ O(\eta), \tag{6}$$

where path kernel $K_t^{P_t}(x, x') = \big\langle \nabla_\Theta f(x, Z_t),\, \nabla_\Theta f(x', Z_t) \big\rangle_{P_t^{-1}}$ and

$$P_t = \operatorname{diag}\left( \mu \int_0^t e^{-\mu(t-s)} \left[ (\nabla L(Z_s))^2 + \operatorname{diag}\big(\Sigma_{|\mathcal{B}|}(Z_s)\big) \right] \mathrm{d}s \right) + \varepsilon\, \mathbf{I}_d,$$

$$\Sigma_{|\mathcal{B}|}(Z) = \mathbb{E}_{\mathcal{B}}\big[ \big(\nabla L_{\mathcal{B}}(Z) - \nabla L(Z)\big) \big(\nabla L_{\mathcal{B}}(Z) - \nabla L(Z)\big)^\top \big].$$

# Stochastic Domingos' Theorem (Adam)

Consider a learning model $y = f(x, \Theta)$. Assume that $\Theta$ is learned from the dataset $\{(x_n, y_n^*)\}_{n=1}^N$ by Adam with learning rate $\eta$, first-moment coefficient $\beta_1 \in [0, 1)$, second-moment coefficient $\beta_2 \in [0, 1)$, and smoothing constant $\varepsilon > 0$. Let $c_1 = (1 - \beta_1)/\eta$, $c_2 = (1 - \beta_2)/\eta$, and suppose that $c_1 = O(\eta^{-\xi})$ for some $\xi \in (0, 1)$ and $c_2 = O(1)$ as $\eta \to 0$. Then

$$\mathbb{E}[f(x, \Theta_k)] = f(x, \Theta_0) - c_1\beta_1 \frac{1}{N} \sum_{n=1}^{N} \mathbb{E}\left[ \int_0^{k\eta} \int_0^t K_{t,s}^{P_t}(x, x_n) \frac{\partial \ell}{\partial f}\big(f(x_n, Z_s), y_n^*\big)\, \mathrm{d}s\, \mathrm{d}t \right]$$

$$- c_1(1 - \beta_1) \frac{1}{N} \sum_{n=1}^{N} \mathbb{E}\left[ \int_0^{k\eta} K_{t,t}^{P_t}(x, x_n) \frac{\partial \ell}{\partial f}\big(f(x_n, Z_t), y_n^*\big)\, \mathrm{d}t \right] + O(\sqrt{\eta}), \tag{7}$$

where stochastic path kernel

$$K_{t,s}^{P_t}(x, x') = \langle \nabla_\Theta f(x, \Theta_t), \frac{\sqrt{1 - e^{-c_2 t}}}{1 - e^{-c_1 t}} e^{-c_1(t-s)} \nabla_\Theta f(x', \Theta_t) \rangle_{P_t^{-1}}$$

and $P_t = c_2^{1/2} \operatorname{diag}\Big( \int_0^t e^{-c_2(t-s)} \big[ \operatorname{diag}\big(\Sigma_{|\mathcal{B}|}(Z_s)\big) + (\nabla L(Z_s))^2 \big]\, \mathrm{d}s \Big)^{1/2} + \varepsilon\sqrt{1 - e^{-c_2 t}}\, \mathbf{I}_d$.

# Evolution of gradient kernel during training

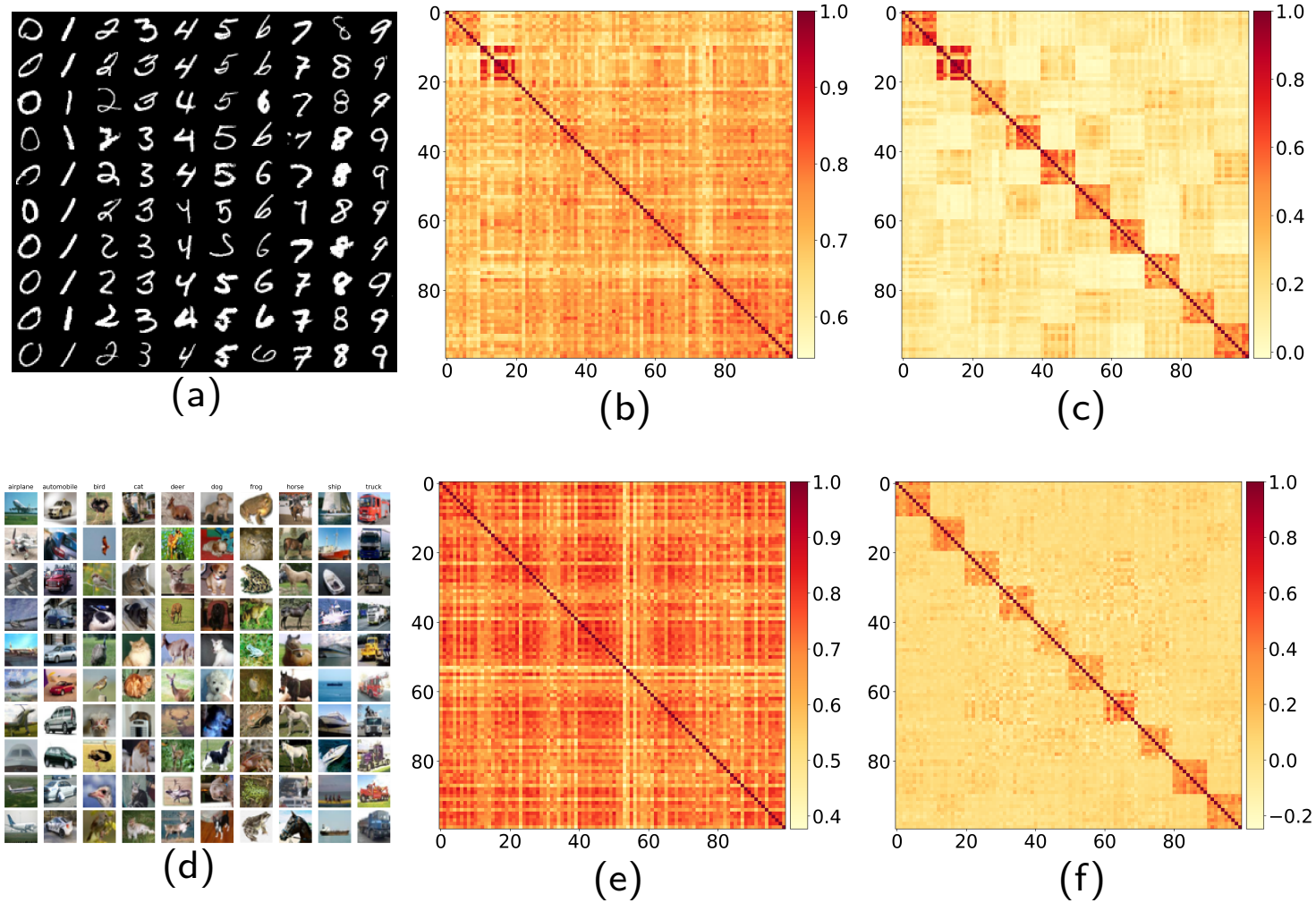



Figure: Normalized path kernel $\widehat{K}_t(x_i, x_j) := K_t(x_i, x_j)/\sqrt{K_t(x_i, x_i)\,K_t(x_j, x_j)}$ on MNIST and CIFAR-10. (a,d) Example samples. (b,e) Kernels at initialization. (c,f) Kernels after training.

# Extrapolation failure in the gradient kernel view

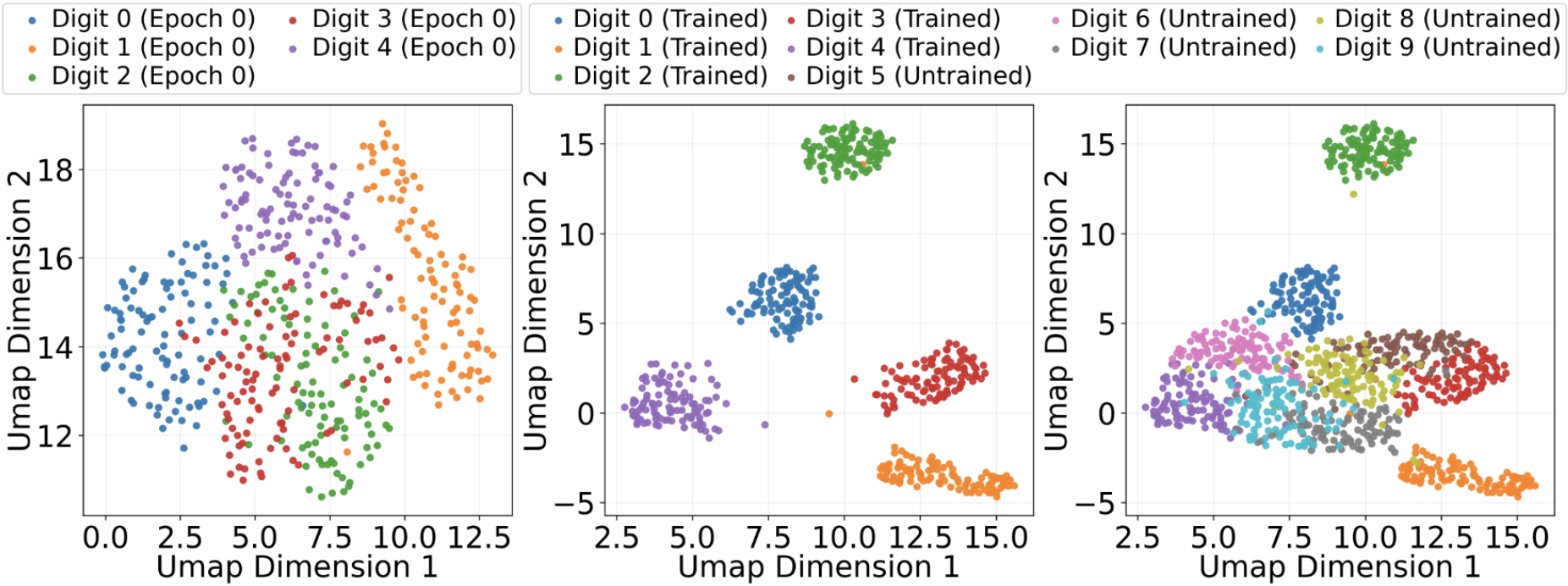



Figure: UMAP visualization of MNIST digits in the tangent feature space $\{\nabla_\Theta f(x_n, \Theta_k)\}_{n=1}^N$. **Left:** Initialization (0-th epoch) on digits 0–4. **Middle:** After training on digits 0–4 (200-th epoch). **Right:** Tangent-feature visualization of all digits: while trained digits (0–4) remain well separated, unseen digits (5–9) fail to form distinct clusters.

# Extrapolation failure in the gradient kernel view

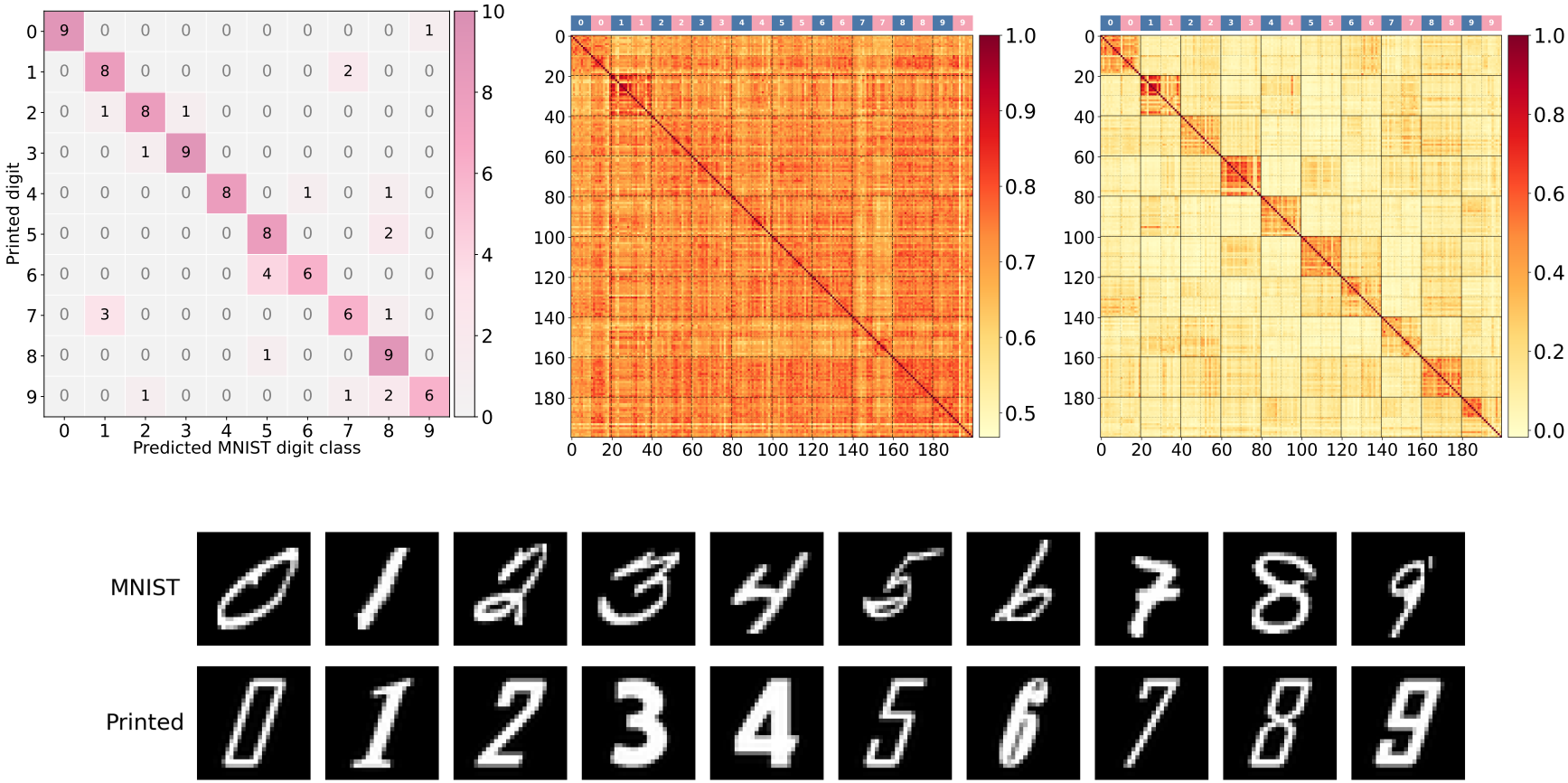
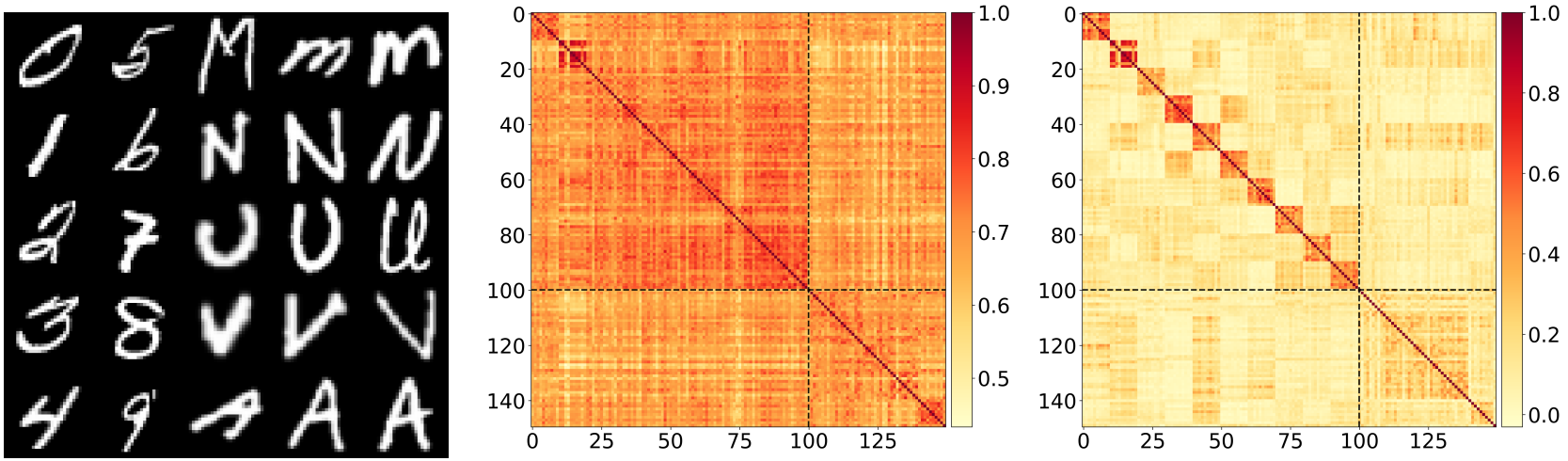



Figure: Normalized gradient kernel for out-of-domain printed digits. Top: normalized gradient kernels at initialization (a) and after training (b), and the printed-digit confusion matrix (c). Bottom: MNIST and printed digit examples. Blue and pink bars indicate MNIST and printed blocks, respectively.

# Extrapolation failure in the gradient kernel view




(a) (b) (c)

Figure: Normalized gradient kernel for out-of-domain inputs (unseen letters). Left: MNIST digits and unseen handwritten letters $M, N, U, V, A$. Middle and right: normalized gradient kernels at initialization and after training.

# RKHS reminder

A positive definite kernel $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ generates a Reproducing kernel Hilbert space (RKHS) $\mathcal{H}_K$ with inner product

$$\langle K(x, \cdot), K(y, \cdot) \rangle_{\mathcal{H}_K} = K(x, y).$$

For every $f \in \mathcal{H}_k$, the reproducing property gives:

$$f(x) = \langle f, K(x, \cdot) \rangle_{\mathcal{H}_K}, \forall x \in \mathcal{X}.$$

Given a data distribution $\mu$, $x' \sim \mu$, define the integral kernel operator

$$[T_K g](x) = \int_{\mathcal{X}} K(x, x') g(x') \, d\mu(x').$$

If $g(x')$ represents the training residual (correction) at point $x'$, then $K(x, x')g(x')$ measures how much the correction at $x'$ contributes to the prediction at $x$.
By integrating over all $x'$, we obtain the total correction received at $x$.

# Generalization Error and Kernel Null Space

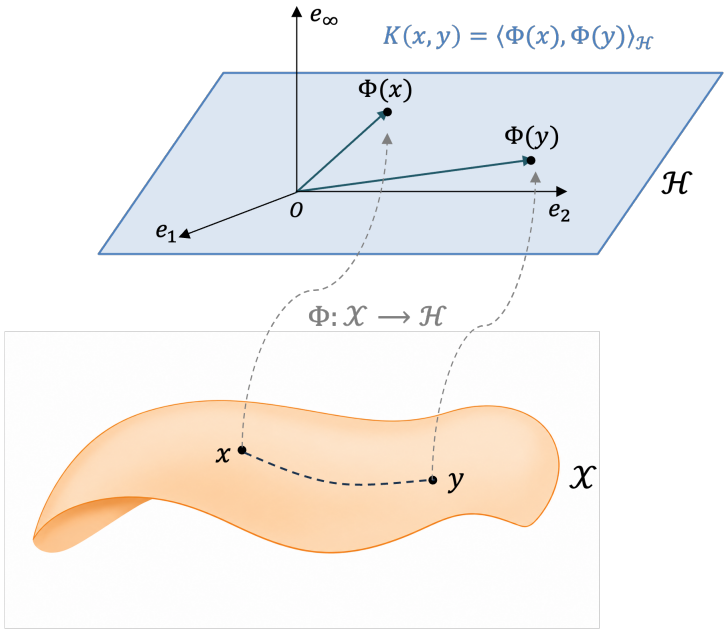
Let $\mathcal{K}_t$ be the integral operator induced by stochastic path kernel $K_t(x, x') = \langle \nabla_\Theta f(x, \Theta_t), \nabla_\Theta f(x', \Theta_t) \rangle$, defined by

$$[\mathcal{K}_t g](x) = \int_{\mathbb{R}^p} K_t(x, x')\, g(x')\, \mathrm{d}\mathcal{P}(x'),$$

where $\mathcal{P}$ is the marginal distribution of $x$ on $\mathbb{R}^p$. The stochastic path kernel induces a trajectory-dependent RKHS

$$\mathcal{H}_t := \mathcal{H}_{K_t}.$$

Mercer's theorem gives $K_t(x, x') = \sum_{i=1}^{\infty} \lambda_i\, \phi_i(x)\, \phi_i(x')$ and $\ker(\mathcal{K}_t) = \mathrm{span}\{\phi_i : \lambda_i = 0\}$ collects the unlearnable directions.



# Generalization Error and Kernel Null Space

**Theorem** (Informal) Under mild assumptions, let $f^*$ be the target function and suppose the loss is the mean-squared error. Define the terminal residual $\Delta_T(x) := f^*(x) - f(x, Z_T)$ and the generalisation error

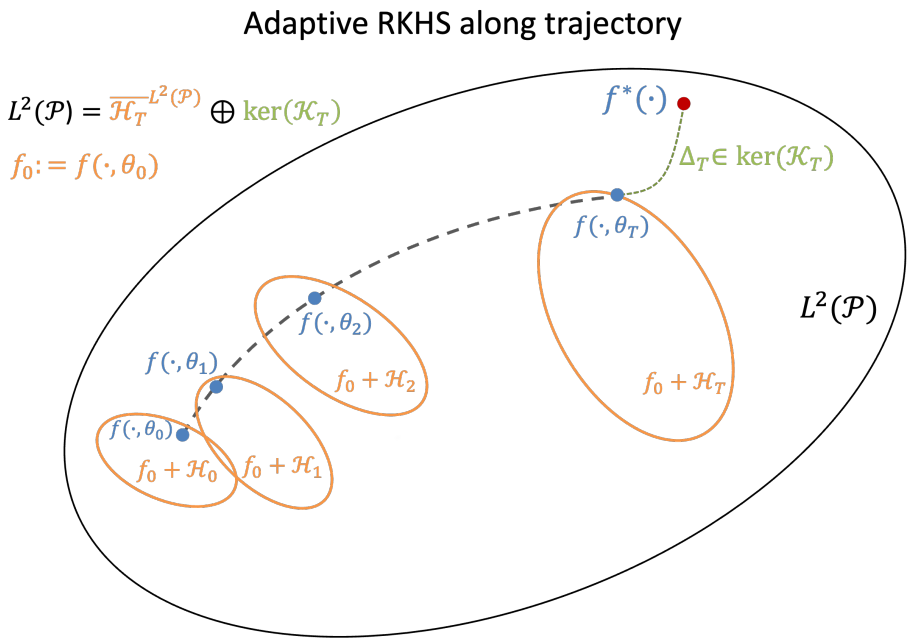
$$\mathcal{E}_T := \|\Delta_T\|^2_{L^2(\mathbb{R}^p,\, \mathcal{P})} = \int_{\mathbb{R}^p} \big(f^*(x) - f(x, Z_T)\big)^2 \, \mathrm{d}\mathcal{P}(x).$$

Then, as $\eta \to 0$,

$$\forall\, \varepsilon > 0, \qquad \mathbb{P}\big(\Delta_T \in \mathcal{N}_\varepsilon\big(\ker(\mathcal{K}_T)\big)\big) \to 1,$$

where $\mathcal{N}_\varepsilon(\cdot)$ denotes the $\varepsilon$-neighbourhood in $L^2(\mathbb{R}^p, \mathcal{P})$. In particular, the generalisation error $\mathcal{E}_T$ is governed by the component of the residual lying in $\ker(\mathcal{K}_T)$.

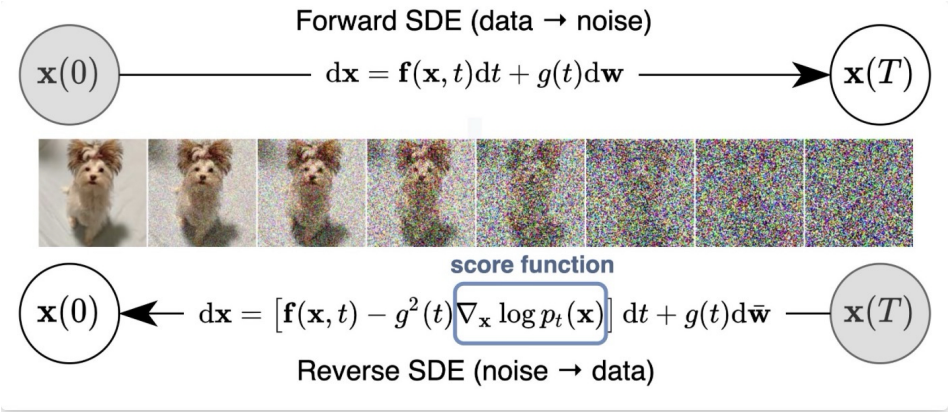
# Generalization Error and Kernel Null Space




At each time $t$, the stochastic path kernel $K_t$ induces an RKHS $\mathcal{H}_t$.

# Path Kernel Perspective for DDPM

(Denoising Diffusion Probabilistic Model)




**Corollary** (Informal) Let $\{(x_{t,n}, \gamma(t)), \varepsilon_t\}_{t=1,n=1}^{T,N}$ denote the training set, where $T$ is the total number of timesteps and $N$ is the number of training samples. Let $\{Z_s\}$ denote the weak first-order approximation of the SGD iterates. Then, for any query $(x, \gamma(\tau))$, the expected noise prediction satisfies

$$\mathbb{E}[\varepsilon_\Theta((x, \gamma(\tau)), \Theta_k)] = \varepsilon_\Theta((x, \gamma(\tau)), \Theta_0)$$

$$- \frac{2}{N \cdot T} \sum_{t=1}^{T} \sum_{n=1}^{N} \mathbb{E}\left[ \int_0^{k\eta} \|\varepsilon_\Theta((x_{t,n}, \gamma(t)), Z_s) - \varepsilon_t\| \, K_s((x, \gamma(\tau)), (x_{t,n}, \gamma(t)))\, ds \right]$$

$$+ O(\eta),$$

where $K_s$ is the stochastic gradient kernel of the noise predictor $\varepsilon_\Theta$.

# Path Kernel Perspective for Generative Models

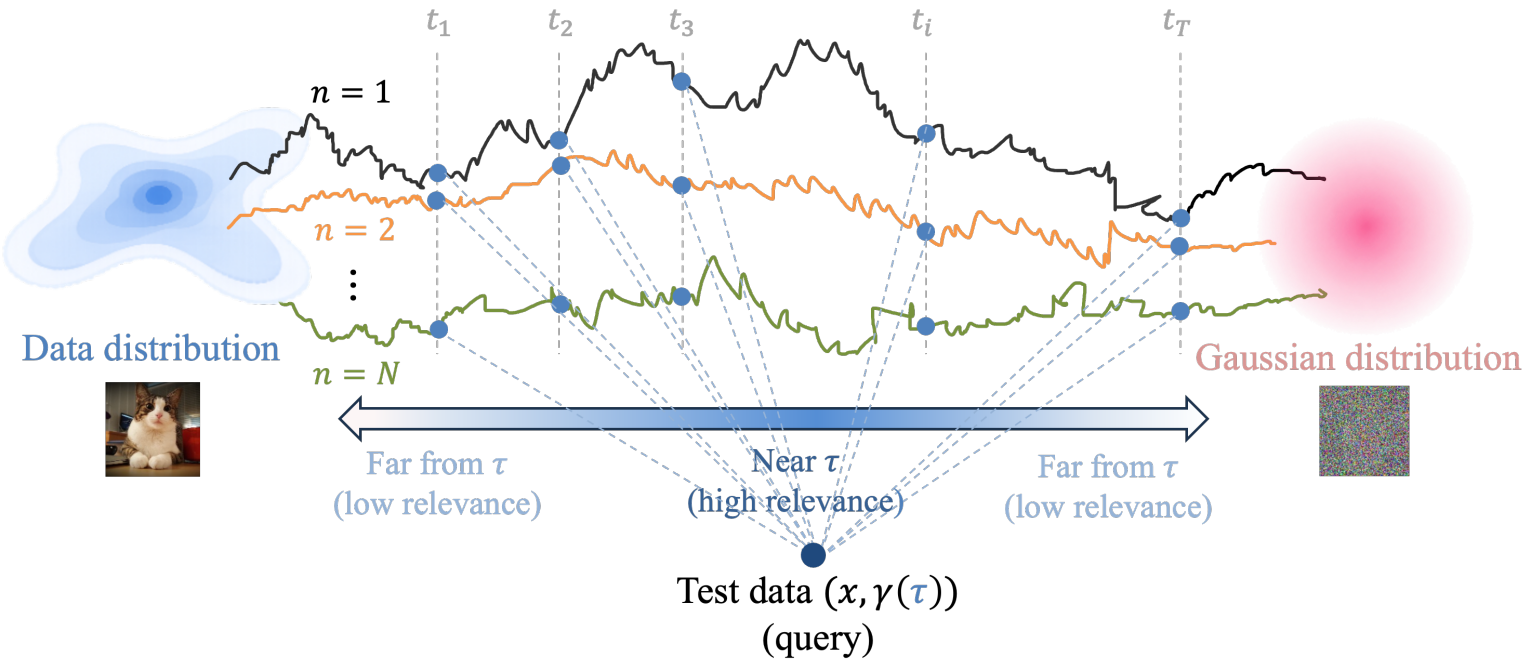
## Path memory over all samples and noise levels

**stored denoising residual**

$$r(t,n,Z_s) := \left\| \varepsilon_\theta\left(\left(x_{t,n},\gamma(t)\right),Z_s\right) - \varepsilon_t \right\|$$

$t \in \{1,2,\cdots,T\}$: noise levels (time partition)

$n \in \{1,2,\cdots,N\}$: trajectories (samples)




Expected generated output

$$\mathbb{E}[\varepsilon_\theta((x,\gamma(\tau)),\theta_k)] = \varepsilon_\theta((x,\gamma(\tau)),\theta_0) - \frac{1}{NT}\sum_{t=1}^{T}\sum_{n=1}^{N}\mathbb{E}\left[\int_0^{k\eta} \underbrace{r(t,n,Z_s)}_{\text{noise-prediction residual}} \underbrace{K_s\left(\left(x,\gamma(\tau)\right),\left(x_{t,n},\gamma(t)\right)\right)}_{\text{interaction weights}} ds\right]$$

$$+O(\eta).$$

# Path kernel perspective for GAN

Generative adversarial networks (GANs) learn a generator $G(\cdot,\cdot) : \mathcal{Z} \times \mathbb{R}^p \to \mathcal{X}$ and a discriminator $D(\cdot) : \mathcal{X} \to (0,1)$ via the minimax objective

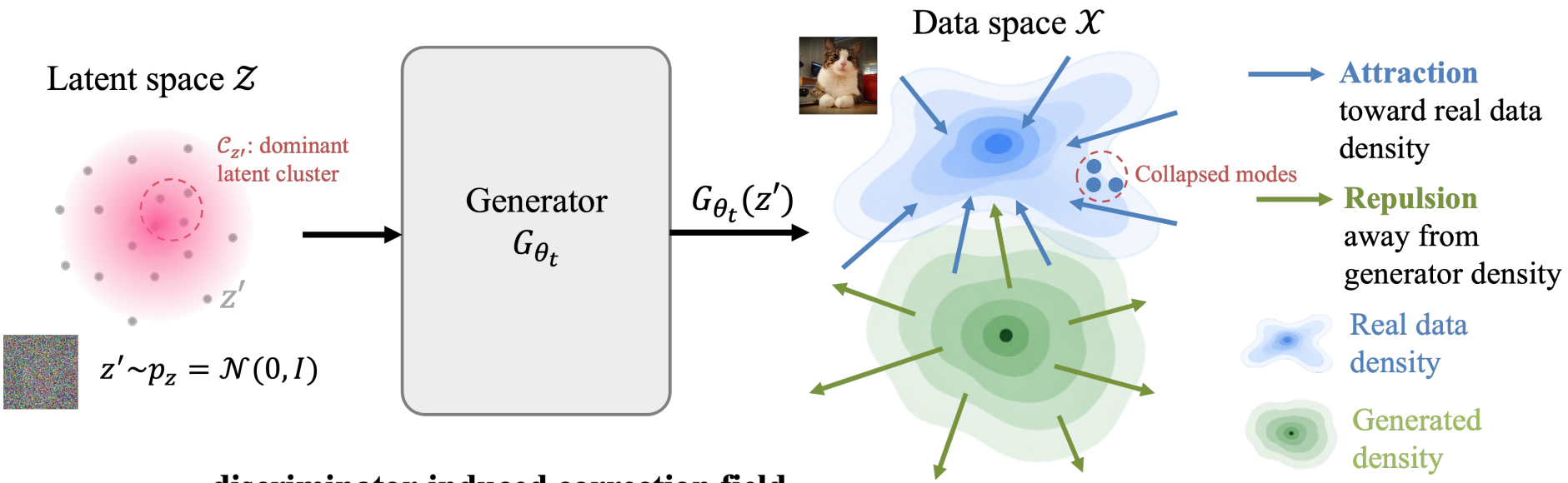
$$\min_{\Theta} \max_{D} \Big\{ \mathbb{E}_{x \sim p_{\text{data}}} \log D(x) + \mathbb{E}_{z \sim p_Z} \log \big(1 - D(G(z,\Theta))\big) \Big\}.$$

**Corollary** (Informal) Consider a generator $G_\Theta := G(\cdot,\Theta)$ trained by SGD with learning rate $\eta$ on latent samples $z' \sim p_Z$. Assume that the discriminator is fully optimized at each iteration and treating $D^*$ as fixed when differentiating with respect to $\Theta$. Let $\{\bar{\Theta}_t\}$ denote the weak first-order approximation of the SGD iterates. Then, when $\nabla_y \log p_{\text{data}}$ and $\nabla_y \log p_{G_t}$ are well-defined, for any query $z$, the expected generator output satisfies

$$\begin{aligned}
\mathbb{E}\big[G_{\Theta_k}(z)\big] &= G_{\Theta_0}(z) - \mathbb{E}\left[\int_0^{k\eta} \mathbb{E}_{z' \sim p(z)} \left[ K_t(z,z') \frac{\partial}{\partial G} \ell\big(D^*(G_{\bar{\Theta}_t}(z'))\big) \right] \mathrm{d}t\right] + O(\eta) \\
&= G_{\Theta_0}(z) + \mathbb{E}\left[\int_0^{k\eta} \mathbb{E}_{z' \sim p(z)} \left[ K_t(z,z') D^*\big(G_{\bar{\Theta}_t}(z')\big) \left(\nabla_y \log p_{\text{data}}(y) - \nabla_y \log p_{G_{\bar{\Theta}_t}}(y)\right)\big|_{y = G_{\bar{\Theta}_t}(z')} \right] \mathrm{d}t\right] \\
&\quad + O(\eta), \qquad (8)
\end{aligned}$$

where $K_t$ is the stochastic gradient kernel of the generator $G_{\bar{\Theta}_t}$, and $\ell(D(G_{\bar{\Theta}_t}(z))) = \log(1 - D(G_{\bar{\Theta}_t}(z)))$.

# Path Kernel Perspective for GAN




**discriminator-induced correction field**

$$v_t(z', \theta_t) := D^*\left(G_{\theta_t}(z')\right)\left(\nabla_y \log p_{data}(y) - \nabla_y \log p_{G_{\theta_t}}(y)\right)\Big|_{y = G_{\bar{\theta}_t}(z')}$$

Expected generated output

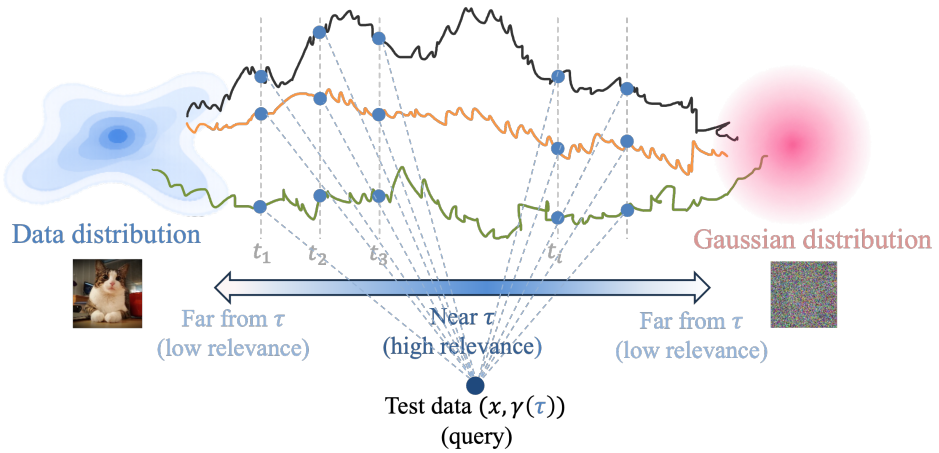
$$\mathbb{E}[G(z, \theta_k)] = G(z, \theta_0) + \mathbb{E}\int_0^{k\eta} \mathbb{E}_{z' \sim p(z)}[\underbrace{K_t(z, z')}_{\substack{\text{interaction} \\ \textbf{weights}}} \underbrace{v_t(z', \theta_t)}_{\substack{\text{correction} \\ \text{field}}}]dt + O(\eta)$$

Mode collapse amplification: A subset of latent samples $\mathcal{C}_{z'}$ dominate the correction field $v_t(z', \theta_t)$.

$\Longrightarrow$ Mode collapse is reinforced during training.

# Path Kernel Correction for Generative Model

Diffusion model




**DM's correction(residual):**

$$r(t, n, Z_s) := \left\| \varepsilon_\theta \left( \left( x_{t,n}, \gamma(t) \right), Z_s \right) - \varepsilon_t \right\|$$

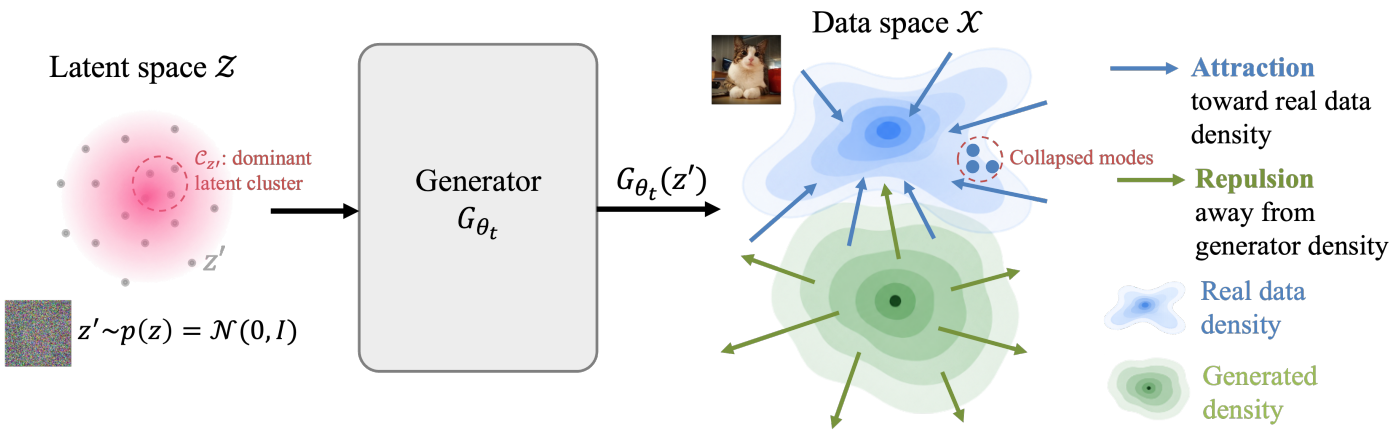
**GAN's correction**

$$v_t(z', \theta_t) := D^* \left( G_{\theta_t}(z') \right) \left( \nabla_y \log p_{data}(y) - \nabla_y \log p_{G_{\theta_t}}(y) \right) \Big|_{y = G_{\theta_t}(z')}$$

Difference:
DM retrieves supervised residuals from a full noise-indexed memory.

GAN propagates distribution-mismatch corrections from the current generated support.

GAN



# What Is Missing from First-Order Path Kernels?

- Domingos' formula gives a first-order explanation of prediction: prediction=initialization+accumulated tangent feature interpolation.

- However, practical training is not purely first-order.
  - The finite learning rate, the curvature of the loss landscape, and the randomness of mini batch sampling introduce higher-order corrections to the prediction.

Can we make these corrections explicit in the interpolation formula? Does a higher-order description still preserve the interpolation structure?

# Second-Order Interpolation: Gradient Descent

**Theorem (Informal)** Consider a differentiable model $f(\cdot, \Theta)$ trained on a set $\{(x_n, y_n^*)\}_{n=1}^N$ via gradient descent with learning rate $\eta > 0$. The model output satisfies

$$f(x, \Theta_K) = f(x, \Theta_0) - \frac{1}{N} \sum_{n=1}^{N} \int_0^T \underbrace{K_t(x, x_n)}_{\text{path kernel}} \underbrace{\frac{\partial \ell}{\partial f}(f(x_n, \Theta_t), y_n^*)}_{\text{training correction}} \, \mathrm{d}t$$

$$- \frac{\eta}{2N} \sum_{n=1}^{N} \int_0^T \underbrace{K_t^{\mathrm{cur}}(x, x_n)}_{\text{curvature-induced kernel}} \underbrace{\frac{\partial \ell}{\partial f}(f(x_n, \Theta_t), y_n^*)}_{\text{training correction}} \, \mathrm{d}t + O(\eta^2),$$

where path kernel $K_t(x, x') := \langle \nabla_\Theta f(x, \Theta_t), \nabla_\Theta f(x', \Theta_t) \rangle$, curvature-induced kernel $K_t^{\mathrm{cur}}(x, x') = \langle \nabla_\Theta f(x, \Theta_t), \nabla_\Theta f(x', \Theta_t) \rangle_{\nabla_\Theta^2 L(\Theta_t)}$.

**Remark**. Under a learning-rate schedule $\eta_k = \eta_{\min} + \frac{1}{2}(\eta_{\max} - \eta_{\min})\Big(1 + \cos\big(\frac{k}{K}\pi\big)\Big)$, $k = 0, \ldots, K$, the prediction satisfies

$$f(x, \Theta_K) = f(x, \Theta_0) - \frac{1}{N} \sum_{n=1}^{N} \int_0^T \Big(w(t) - \frac{\eta}{2}\dot{w}(t)\Big) K_t(x, x_n) \frac{\partial \ell}{\partial f}\big(f(x_n, \Theta_t), y_n^*\big) \, dt$$

$$- \frac{\eta}{2N} \sum_{n=1}^{N} \int_0^T w(t)^2 K_t^{\mathrm{cur}}(x, x_n) \frac{\partial \ell}{\partial f}\big(f(x_n, \Theta_t), y_n^*\big) dt + O(\eta^2),$$

with the scalar weight $w(t) = 1 + \frac{1}{2}(\kappa - 1)\big(1 + \cos(\frac{\pi t}{T})\big)$, $\kappa = \eta_{\max}/\eta_{\min}$, $t \in [0, T]$.

# Second-Order Interpolation: Gradient Descent

**Proof.** Let $\Theta_t \in \mathbb{R}^d$ denote the parameter trajectory of the model $f(\cdot, \Theta)$. Following the standard modified equation for gradient descent, we approximate the discrete update $\Theta_{k+1} = \Theta_k - \eta \nabla L(\Theta_k)$ by continuous-time dynamics

$$\dot{\Theta}_t = -\nabla_\Theta L(\Theta_t) - \frac{\eta}{2} \nabla_\Theta^2 L(\Theta_t) \nabla_\Theta L(\Theta_t) + O(\eta^2). \tag{9}$$

By the chain rule and (9), we obtain

$$\begin{aligned}
\frac{\mathrm{d}f}{\mathrm{d}t}(x, \Theta_t) &= \nabla_\Theta f(x, \Theta_t)^\top \frac{\mathrm{d}\Theta_t}{\mathrm{d}t} \\
&= \nabla_\Theta f(x, \Theta_t)^\top \left( -\nabla_\Theta L(\Theta_t) - \frac{\eta}{2} \nabla_\Theta^2 L(\Theta_t) \nabla_\Theta L(\Theta_t) + O(\eta^2) \right) \\
&= -\frac{1}{N} \sum_{n=1}^{N} \langle \nabla_\Theta f(x, \Theta_t), \nabla_\Theta f(x_n, \Theta_t) \rangle \frac{\partial \ell}{\partial f}(f(x_n, \Theta_t), y_n^*) \\
&\quad - \frac{\eta}{2N} \sum_{n=1}^{N} \langle \nabla_\Theta f(x, \Theta_t), \nabla_\Theta f(x_n, \Theta_t) \rangle_{\nabla_\Theta^2 L(\Theta_t)} \frac{\partial \ell}{\partial f}(f(x_n, \Theta_t), y_n^*) + O(\eta^2),
\end{aligned}$$

where $\langle \nabla_\Theta f(x, \Theta_t), \nabla_\Theta f(x_n, \Theta_t) \rangle_{\nabla_\Theta^2 L(\Theta_t)} = \nabla_\Theta f(x, \Theta_t)^\top \nabla_\Theta^2 L(\Theta_t) \nabla_\Theta f(x_n, \Theta_t)$.
Integrating both sides from 0 to $T$ yields the result.

# Second-Order Interpolation: SGD

Compared with GD, SGD introduces an additional source of randomness: the mini-batch gradient noise.

**Theorem (Informal)** Consider a learning model $y = f(x, \Theta)$ trained by SGD with learning rate $\eta$. In the sense of a second-order weak approximation of SGD, we have

$$\begin{aligned}\mathbb{E}[f(x,\Theta_K)] = f(x,\Theta_0) &- \frac{1}{N}\sum_{n=1}^{N}\mathbb{E}\left[\int_0^T K_s(x,x_n)\frac{\partial \ell}{\partial f}\left(f(x_n,Z_s),y_n^*\right)ds\right] \\ &- \frac{\eta}{2N}\sum_{n=1}^{N}\mathbb{E}\left[\int_0^T K_s^{\text{cur}}(x,x_n)\frac{\partial \ell}{\partial f}\left(f(x_n,Z_s),y_n^*\right)ds\right] \\ &+ \frac{\eta}{2}\mathbb{E}\left[\int_0^T \underbrace{\text{Tr}\left(\nabla_\Theta^2 f(x,Z_s)\,\Sigma_{|\mathcal{B}|}(Z_s)\right)}_{\text{sampling-induced term}} ds\right] + O(\eta^2),\end{aligned}$$

where $K_t(x,x') := \left\langle \nabla_\Theta f(x,\Theta_t),\, \nabla_\Theta f(x',\Theta_t)\right\rangle$, curvature-induced kernel $K_t^{\text{cur}}(x,x') = \left\langle \nabla_\Theta f(x,\Theta_t),\, \nabla_\Theta f(x',\Theta_t)\right\rangle_{\nabla_\Theta^2 L(\Theta_t)}$.

This sampling-induced term is specific to SGD and depends on the mini-batch sampling noise, it reflects the interaction between mini-batch randomness and Hessian of the model output.

# Second-Order Interpolation: SGDM

**Theorem (Informal)** Consider a learning model $y = f(x, \Theta)$ learned from a dataset $\{(x_n, y_n^*)\}_{n=1}^N$ using stochastic gradient descent with momentum (SGDM). In the sense of a second-order weak approximation of SGDM, we have

$$\mathbb{E}[f(x, \Theta_K)] = f(x, \Theta_0) - \frac{1}{N}\sum_{n=1}^{N} \mathbb{E}\left[\int_0^T \int_0^t K_{t,s}(x, x_n) \frac{\partial \ell}{\partial f}\left(f(x_n, Z_s), y_n^*\right) ds\, dt\right]$$

$$- \frac{\eta}{2N}\sum_{n=1}^{N} \mathbb{E}\left[\int_0^T \int_0^t K_{t,s}^{\mathrm{cur}}(x, x_n) \frac{\partial \ell}{\partial f}\left(f(x_n, Z_s), y_n^*\right) ds\, dt\right]$$

$$- \frac{\eta}{2N}\sum_{n=1}^{N} \mathbb{E}\left[\int_0^T K_t(x, x_n) \frac{\partial \ell}{\partial f}\left(f(x_n, Z_t), y_n^*\right) dt\right]$$

$$+ \sqrt{\eta}\, \mathbb{E}\left[\int_0^T \int_0^t \underbrace{e^{-\mu(t-s)} \operatorname{Tr}\left((D_s Z_t)^\top \nabla_\Theta^2 f(x, Z_t)\, \Sigma_{|\mathcal{B}|}^{1/2}(Z_s)\right) ds}_{\text{sampling-induced term}}\, dt\right] + O(\eta^2),$$

where $\mu = \frac{1-\beta}{\eta}$ is fixed, $\beta$ is momentum weight, and

$$K_{t,s}(x, x_n) := \left\langle \nabla_\Theta f(x, Z_t),\, e^{-\mu(t-s)} \left(1 - \frac{\mu^2}{2}(t-s)\eta\right) \nabla_\Theta f(x_n, Z_s) \right\rangle,$$

$$K_{t,s}^{\mathrm{cur}}(x, x_n) := \left\langle \nabla_\Theta f(x, Z_t),\, e^{-\mu(t-s)} \nabla_\Theta f(x_n, Z_s) \right\rangle_{\int_s^t \nabla_\Theta^2 L(Z_r)\, dr}.$$

Here, $D_s$ denotes the Malliavin derivative with respect to the driving Brownian motion at time $s$.

# Fluctuation around the interpolation formula

Higher-order terms still preserve the interpolation structure.
They refine it by adding curvature-weighted and sampling-induced interpolation corrections.

The formula gives the mean prediction, but how far is the realized prediction from it?

A simplified form of Proposition 4.1[3] gives, for any $\delta \in (0,1)$,

$$\mathbb{P}\left( |f(x,\Theta_K) - \mathbb{E}[f(x,\Theta_K)]| \le S_{|\mathcal{B}|}\, G_f(x) \sqrt{\eta\, \frac{e^{2\widetilde{K}_\eta T} - 1}{\widetilde{K}_\eta} \log\frac{2}{\delta}} \right) \ge 1 - \delta.$$

Here $S_{|\mathcal{B}|}$ is an upper bound for the mini-batch noise covariance, $G_f(x) > 0$ bounds $\|\nabla_\Theta f(x,z)\|$ on $\mathcal{U}$, and $\widetilde{K}_\eta$ collects the regularity constants of the drift and diffusion terms.

[3] Jin Guo, Roy Y. He, and Jean-Michel Morel. Second-Order Path Kernel Interpolation Formulas in Machine Learning. arXiv preprint arXiv:2606.07495, 2026.

# Scaling order verification

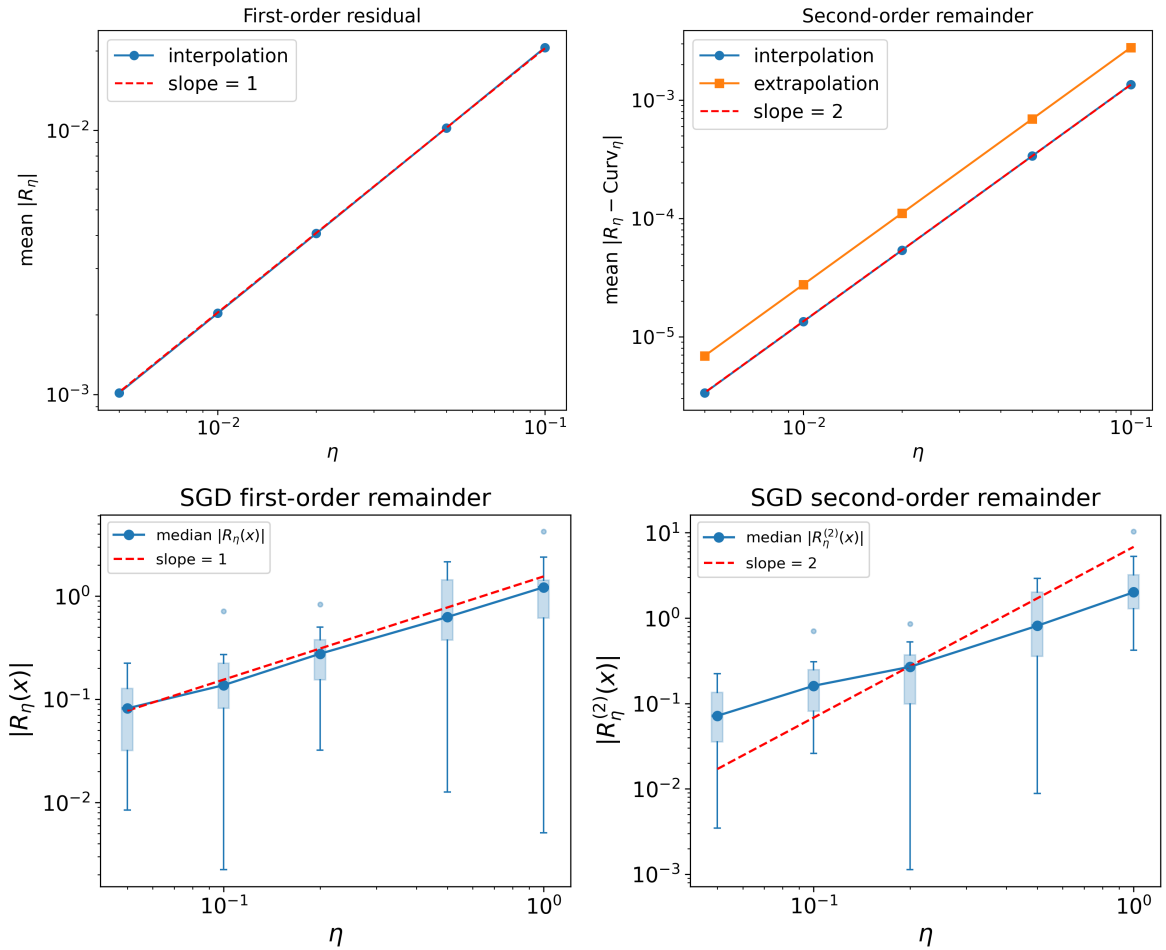



Figure: Scaling verification of the output expansion. Top: GD remainders show first-order $O(\eta)$ and second-order $O(\eta^2)$ scaling after subtracting the corresponding path-kernel and curvature terms. Bottom: SGD remainders show the same empirical scaling across 20 independent runs.

# Summary

- **Training as memory.** Neural networks store residuals along the parameter path. Predictions are obtained by retrieving these residuals through path-kernel interaction weights. Thus, the network prediction is a trajectory-dependent interpolation of the training data.
- **Stochasticity changes the memory weights.** Mini-batch noise does not destroy the path-kernel representation. Different optimizers change the weights of training corrections.
- **Generalization through adaptive RKHS.** The stochastic path kernel induces an adaptive RKHS, and the generalization error lies in the null space of the associated kernel operator.
- **Generative models through the same lens.** Diffusion models and GANs both admit a path-kernel correction form.

THANK YOU!

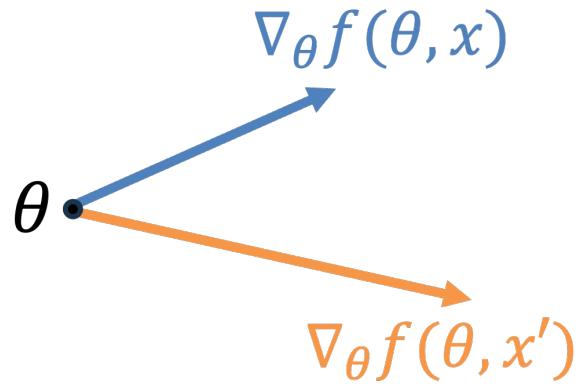
# Neural Tangent Kernel

NTK gives a static kernel view of neural network prediction.

- Neural tangent kernel(NTK)[1]:

$$K_\theta^{\mathrm{NTK}}(x, x') = \langle \nabla_\theta f(\theta, x), \nabla_\theta f(\theta, x') \rangle$$

measures the similarity between data points $x, x'$ by comparing their gradients.




- Under infinite width limit and NTK initialization, the NTK view is static.

$$K_{\theta_0}^{\mathrm{NTK}}(x, x') \approx K_{\theta_\infty}^{\mathrm{NTK}}(x, x').$$

The kernel is (approximately) fixed at initialization.

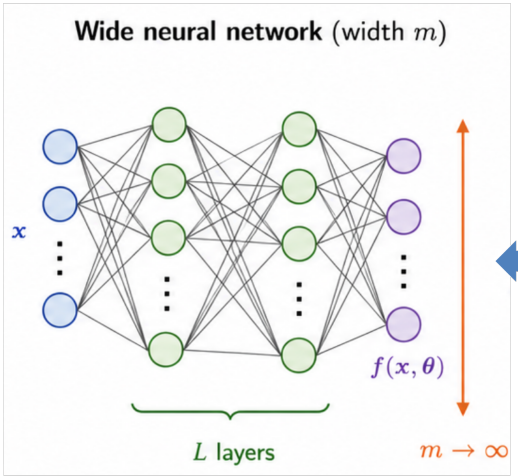
[1] Jacot, A., Gabriel, F. and Hongler, C., 2018. Neural tangent kernel: Convergence and generalization in neural networks. Advances in neural information processing systems, 31.

# Neural Tangent Kernel

- Wide neural networks are linear in the parameter space

$$f(x, \theta_t) = f(x, \theta_0) + \langle \nabla_\theta f(x, \theta_0), \theta_t - \theta_0 \rangle + O\left(\frac{1}{m}\right), m: \text{ width of NN.}$$

The NTK regime views neural network training through a nearly fixed linearized model around initialization.




Tangent feature view
(at initialization $\theta_0$)

$$f(x, \theta_t) - f(x, \theta_0) \approx \Phi(x)^\top (\theta_t - \theta_0),$$

$$\Phi(x) = \nabla_\theta f(x, \theta_0)$$

Useful for analyzing infinite-width NNs.

But finite width training can move beyond this linearized regime.