Feature Learning in Deep Foundation Models: Optimality in Transformers and Diffusion Models

Taiji Suzuki

The University of Tokyo

Deep Learning Theory Team/AIP-RIKEN

4th/May/2026

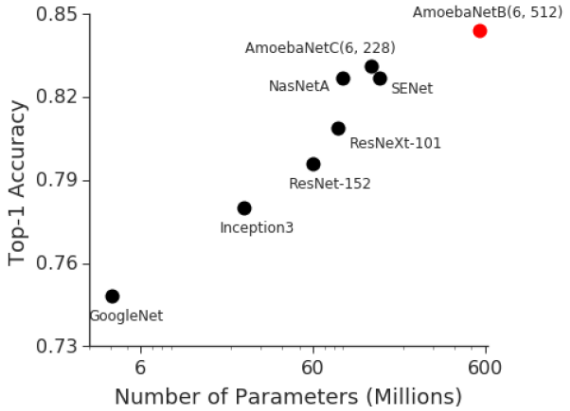
AISTATS2026@Tangier, Morocco

Computational resource for training AI

Computational effort: 2 times larger for 6 months

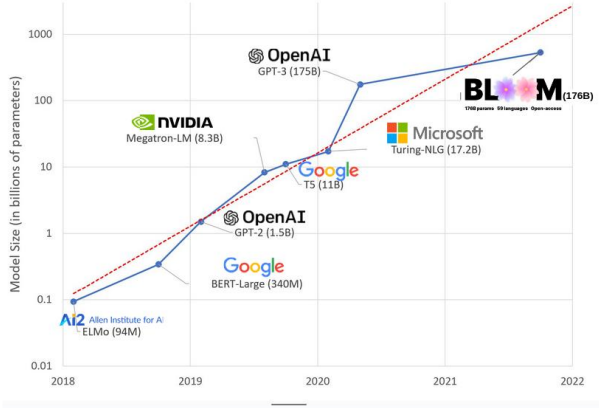
"Computing Power and the Governance of Artificial Intelligence"
Sastry, Heim, Belfield, Anderljung, Brundage, Hazell, O'Keefe, Hadfield et al., 2024

[Sastry et al.: Computing Power and the Governance of Artificial Intelligence. arXiv:2402.08797]



Model size vs performance

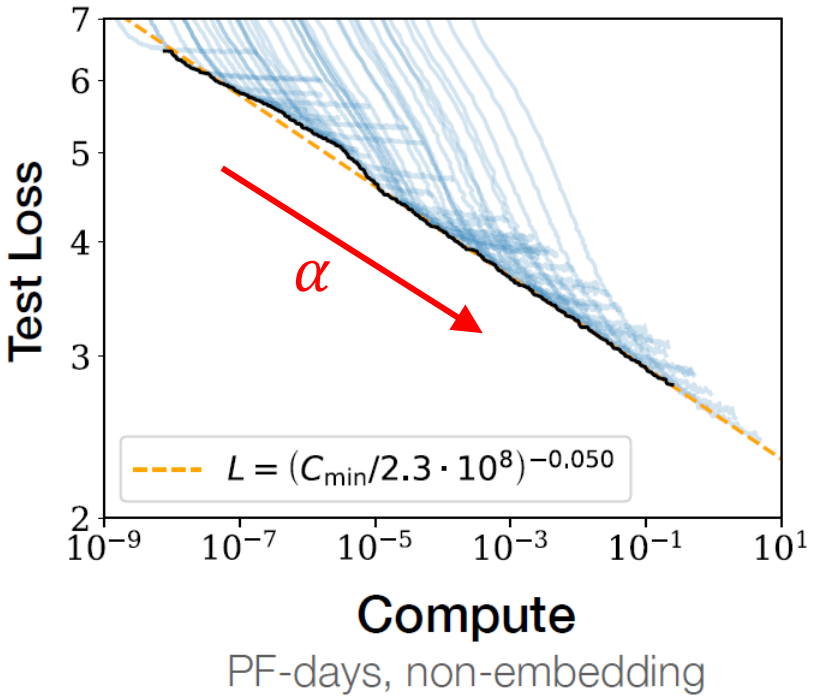
[Real et al.: Regularized evolution for image classifier architecture search. 2019]



Exponential growth of model size

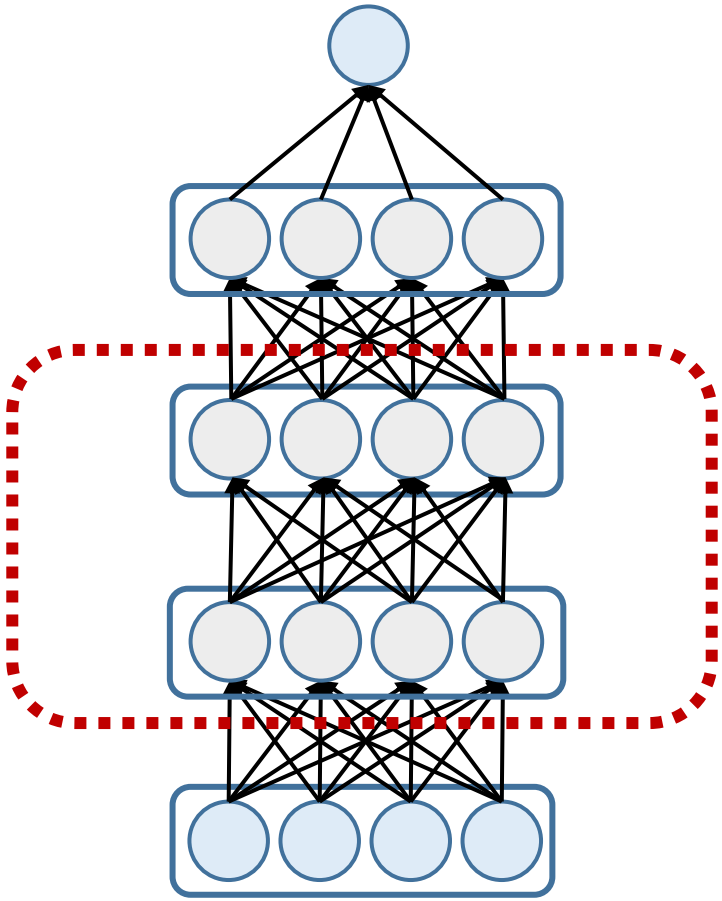
[Luccioni. The mounting human and environmental costs of Generative AI. Apr. 2023.]

Scaling law



[Kaplan et al.: Scaling Laws for Neural Language Models, 2020]

$$\log(\text{Predictive error}) = -\alpha \log(n) + \log(C)$$



- Feature learning (compressed information)
- Data dependent basis functions

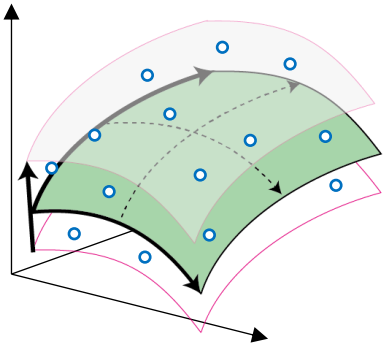
Deep vs Shallow (different scaling law)

Feature learning offers better scaling law.

Low dimensional data structure

[Schmidt-Hieber, 2019] [Nakada&Imaizumi, 2019][Chen et al., 2019][Suzuki&Nitanda, 2019]

If data are distributed on **low dim-manifold**, DL is better.



Deep

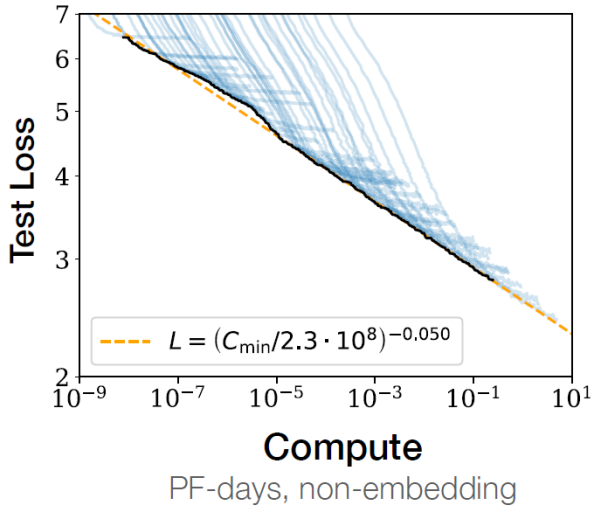
$$n^{-\frac{2s}{2s+D}}$$

Kernel

$$n^{-\frac{2(s-D/p+d/2)}{2(s-D/p+d/2)+d}} \vee n^{-\frac{2s}{2s+D}}$$

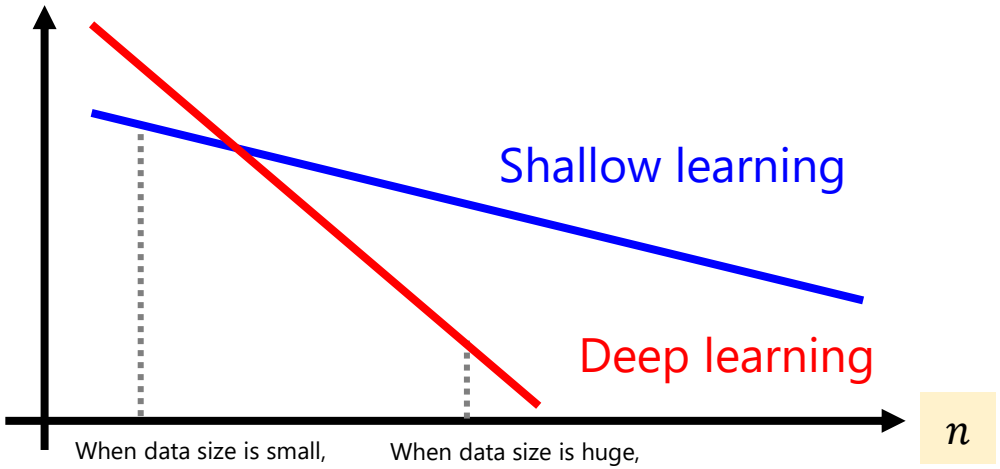
Pred. error

Curse of dimensionality



[Kaplan et al.: Scaling Laws for Neural Language Models, 2020]

$$\log(\text{Predictive error}) = -\alpha \log(n) + \log(C)$$



Feature learning of foundation models

- Transformer

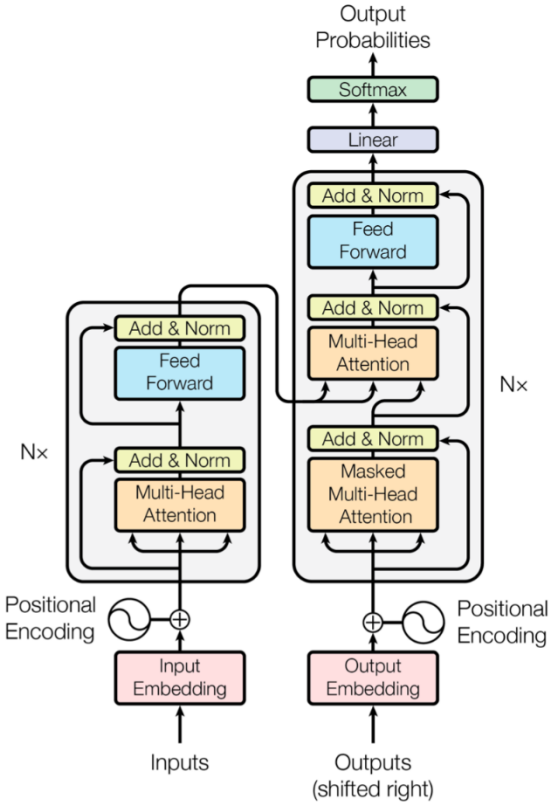
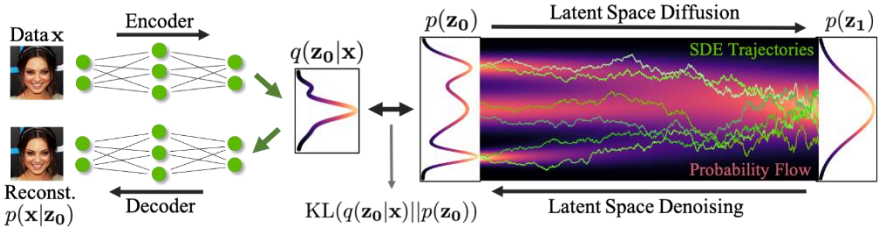


Figure 1: The Transformer - model architecture.

[Vaswani et al.: Attention is All you Need. NIPS2017]

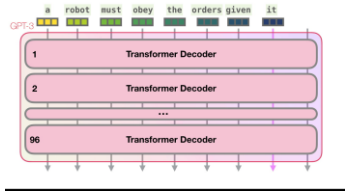
- Diffusion models

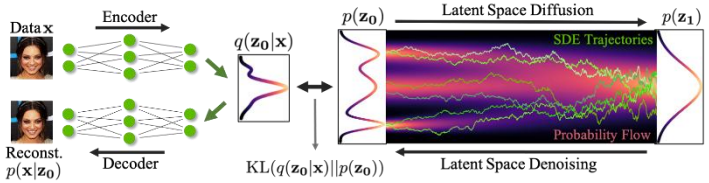
Stable diffusion, 2022.



[Vahdat, Kreis, Kautz: Score-based Generative Modeling in Latent Space. arXiv:2106.05931]

Presentation overview





[Vahdat, Kreis, Kautz: Score-based Generative Modeling in Latent Space. arXiv:2106.05931]

Part 1: In-context learning

Test time feature learning by softmax attention
- Achieving the sample complexity with generative exponent
- Gaussian single index model

Measure theoretic in-context learning
- Infinite-length context
- Measure to real-value problem

Part 2: Diffusion models

Minimax optimality
- Adaptation to smoothness
- Adaptation to low-dim subspace

General data support
- Curved manifold
- Anisotropic smoothness

- [In-context learning] Naoki Nishikawa, Yujin Song, Kazusato Oko, Denny Wu, Taiji Suzuki: Nonlinear transformers can perform inference-time feature learning: a case study of in-context learning on single-index models. ICML2025.
- [Measure theoretic associative recall] Kawata Ryotaro, Taiji Suzuki: Transformers as measure-theoretic associative memory: A statistical perspective and minimax optimality. ICLR2026.
- [Diffusion models] (1) Kazusato Oko, Shunta Akiyama, Taiji Suzuki: Diffusion Models are Minimax Optimal Distribution Estimators. ICML2023. (2) Guoji Fu, Taiji Suzuki, Wee Sun Lee, Atsushi Nitanda: Beyond the OU Process: Dimension-Adaptive Drifts for Minimax OptimalScore-Based Generative Modeling on Low-dimensional Manifolds. 2026. (3) Yuto Maki, Taiji Suzuki: Efficiency of Diffusion Models for Infinite-dimensional Data Generation: Dimension Independent Approximation and Estimation Errors. 2026.

In-context learning

[Naoki Nishikawa, Yujin Song, Kazusato Oko, Denny Wu, Taiji Suzuki: Nonlinear transformers can perform inference-time feature learning: a case study of in-context learning on single-index models. ICML2025]
[Wakayama, Suzuki: In-Context Learning Is Provably Bayesian Inference: A Generalization Theory for Meta-Learning. ICML2026]
[Oko, Song, Suzuki, Wu: Transformer efficiently learns low-dimensional functions in context. NeurIPS2024]

In-context learning

Pretrained Large Language Models (LLMs) have significant ability of In-Context Learning (ICL) [Brown et al., 2020].

context

Please guess the number that fits in the '?'.

1,1 -> 2
2,3 -> 5
8,13 -> 21
6,0 -> 6
10,1 -> 11
5,27 -> ?

Question

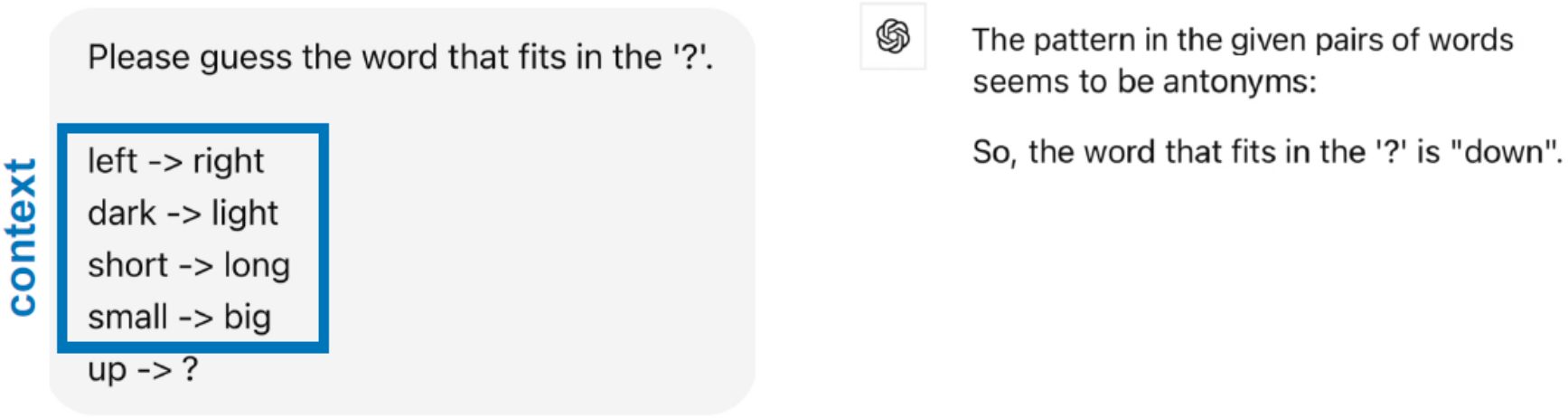
The pattern in the given pairs of numbers appears to be the sum of the two numbers.

So, the number that fits in the '?' is 32.

ChatGPT

In-context learning

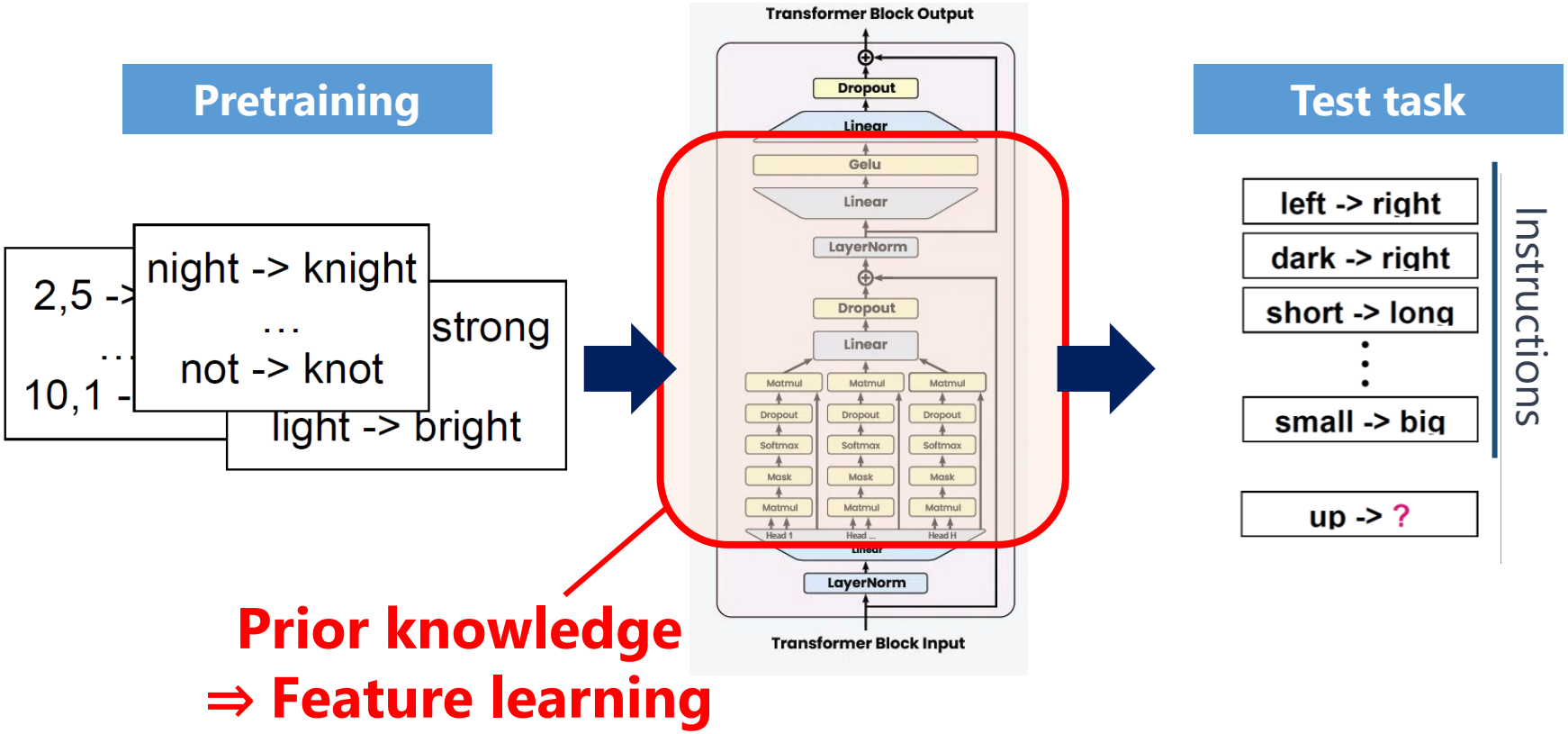
Pretrained Large Language Models (LLMs) have significant ability of In-Context Learning (ICL) [Brown et al., 2020].



Question **ChatGPT**

In-Context learning

ICL is performed **without updating model parameters** unlike the traditional "fine-tuning" regime in the test task.
→ Meta-learning



During pretraining, several tasks are observed to train the model.
That information is stored in the model as prior knowledge.
→ Task generalization.

Mathematical formulation of in-context learning

Model: $y_{i,t} = f_*^t(x_{i,t}) + \epsilon_{i,t} \qquad (i = 1, \ldots, n)$

$t = 1, \ldots, T$: Task index

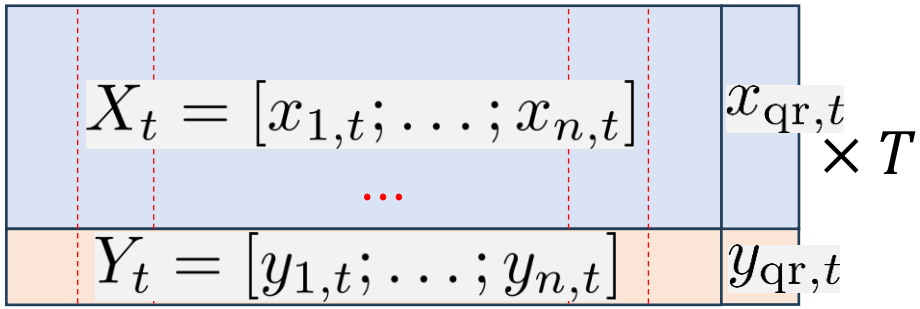
n: number of instructions par task
T: number of pretraining tasks

f_*^t is randomly generated for each task t.

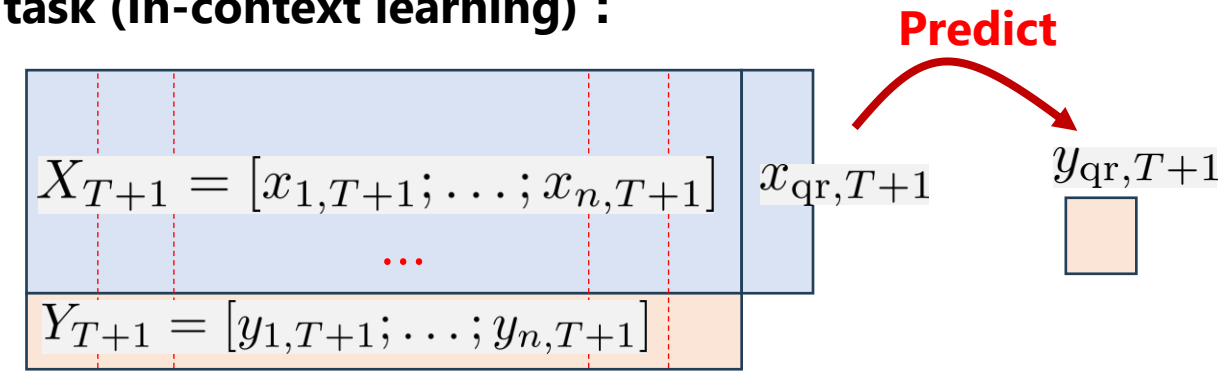
↓

Prior distribution

Pretraining (T tasks) :



Test task (In-context learning) :



ICL risk and Bayes prediction

Expected ICL risk: $D_n = \{(x_1, y_1), \ldots, (x_n, y_n)\}$

$$\mathcal{L}(\Theta) := \mathbb{E}_{f_*}\left[\mathbb{E}_{x_{\mathrm{qr}}, D_n | f_*}\left[\left(f_*(x_{\mathrm{qr}}) - f_\Theta(x_{\mathrm{qr}}; D_n)\right)^2\right]\right]$$

Expectation w.r.t. the true

Expectation w.r.t. data given the true f^*

Minimize the predictive risk of an estimator f_Θ computed by n instructions at test time.

⇒ Learn how to learn : **meta-learning**

"Learning to learn"

The ideal ICL risk minimizer is exactly the **Bayes estimator:**

Marginal distribution of instruction data

$$\mathcal{L}(\Theta) = \mathbb{E}_{D_n}\left[\mathbb{E}_{f_* | D_n}\left[\|f_* - f_\Theta(\cdot; D_n)\|^2_{L^2(P_X)}\right]\right]$$

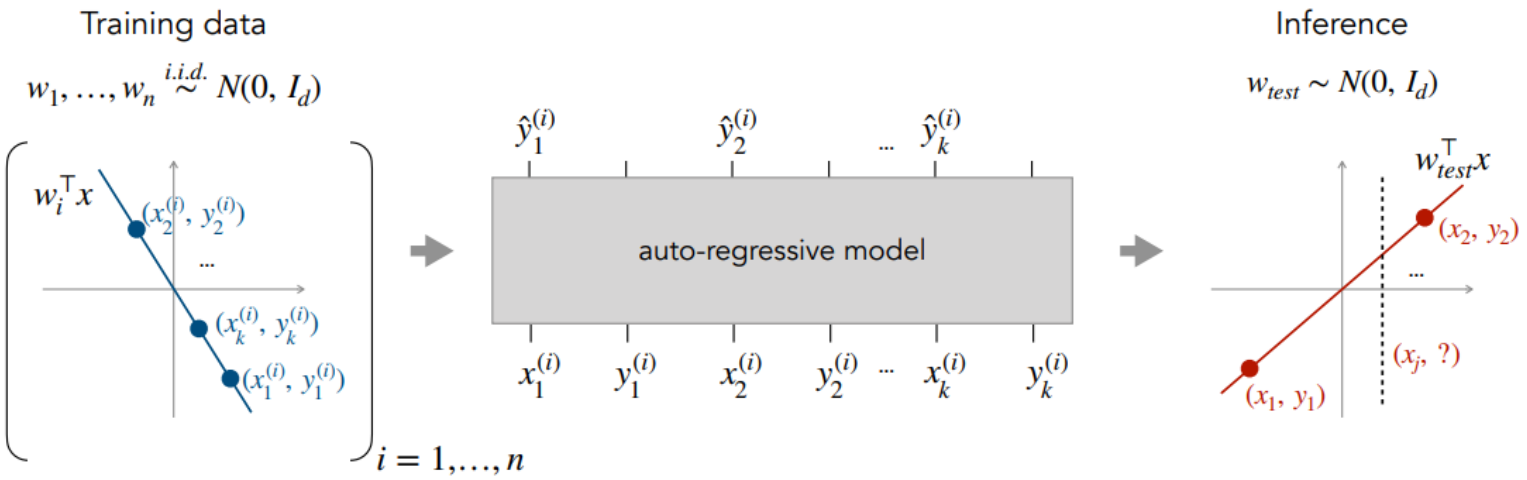
Posterior distribution of f^* given data D_n

$\hat{f}$ minimizes the Bayes posterior mean of loss ⇒ Bayes estimator

How does it produce features in the intermediate layers?

Transformer can implement linear regression

[Gang et al. 2022; Akyurek et al. 2023; Zhang et al. 2023; Ahn et al. 2023; Mahankali et al., 2023; Wu et al. 2024]



[Gang et al.: What Can Transformers Learn In-Context? A Case Study of Simple Function Classes. NeurIPS2022]

Query　KQ matrix　Value・Key

$$\widehat{y}_{\text{query}} = [f_{\text{LSA}}(P; W_*^{KQ}, W_*^{PV})]_{d+1,M+1} = x_{\text{query}}^\top \left(\Lambda + \frac{1}{N}\Lambda + \frac{\text{tr}(\Lambda)}{N} I_d\right)^{-1} \left(\frac{1}{M}\sum_{i=1}^{M} y_i x_i\right)$$

Prompt instructions

[Zhang, Frei, Bartlett.: Trained Transformers Learn Linear Models In-Context. JJMLR, 2024]

Gaussian single index model

Model: $y_{i,t} = f_*^t(x_{i,t}) + \epsilon_{i,t} \qquad (i = 1, \ldots, n)$

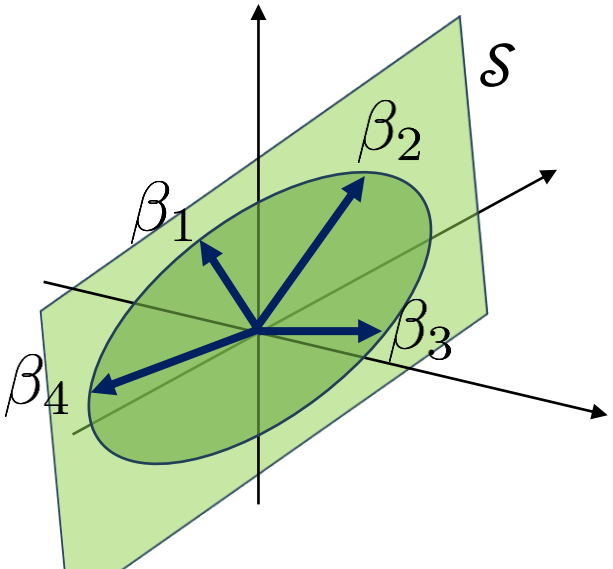
$t = 1, \ldots, T$: Task index

Assumption:

$$x_{i,t} \sim N(0, I)$$
$$f_*^t(x) = \sigma_*(\langle x, \beta_t \rangle)$$

- $\beta_t \sim \mathrm{Unif}(\mathrm{Unit}(\mathcal{S}))$ where $\dim(\mathcal{S}) = r \ll d$

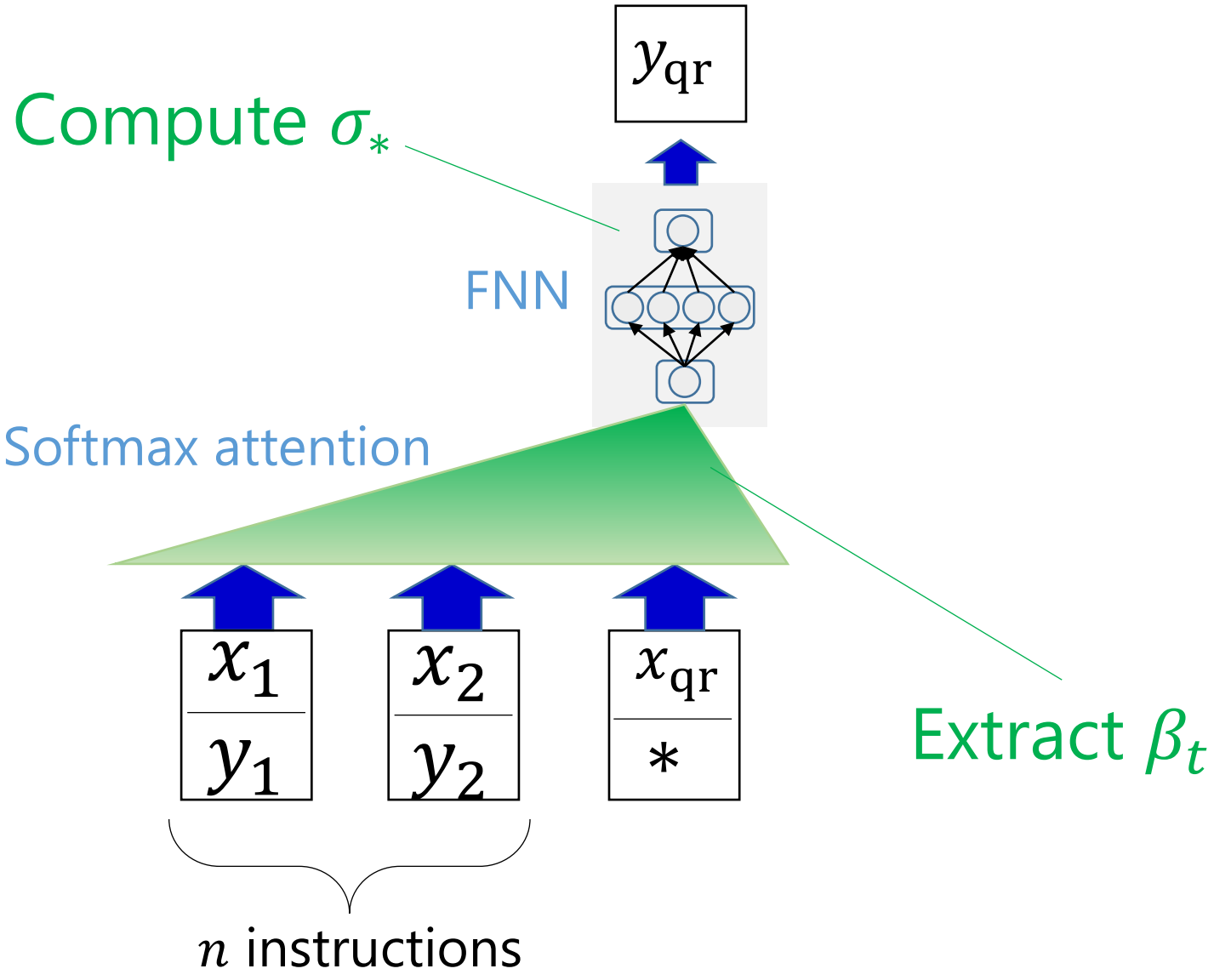
Remark: We assume σ_* is shared across all tasks.



2 ways of feature learning

- **Pretraining:**
  Identify the subspace S
  (Otherwise, performance of test-time inference depends on the ambient dimension d)
- **Test time:**
  Identify β_t from the context.

Model architecture



Condition of the link function

$$f_*(x) = \sigma_*(z) = \sum_{i=k}^{p} \alpha_i^* \mathrm{He}_i(z)$$

He_i: Hermite polynomial with degree i.

Def (Information exponent k [Ben-Arous et al. 2021])

The information exponent of σ_* is the smallest index with non-zero α_i^*:

$$k := \mathrm{IE}(\sigma_*) = \min\{i \in \mathbb{N} \mid \alpha_i^* \neq 0\}$$

Def (Generative exponent [Damian et al. 2024])

The generative exponent of σ_* is the smallest index with non-zero α_i^*:

$$\begin{aligned} \mathrm{ge}(\sigma_*) &= \inf_{\mathcal{T}} \min\{i \in \mathbb{N} \mid \mathbb{E}_{Z\sim N(0,1)}[\mathcal{T} \circ \sigma_*(Z)\mathrm{He}_i(Z)] \neq 0\} \\ &= \inf_{\mathcal{T}} \mathrm{IE}(\mathcal{T} \circ \sigma_*). \end{aligned}$$

SQ, CSQ lower bound

$$\nabla_w \mathbb{E}_{x,y}[(y-\sigma(w^\top x))^2] \propto -\underbrace{\mathbb{E}_{x,y}[y\nabla_w \sigma(w^\top x)]}_{\text{Correlation query}} + \mathbb{E}_x[\sigma(w^\top x)\nabla_w \sigma(w^\top x)]$$

How many (noisy) correlation query should be observed?

- **Correlation statistical query (CSQ):** Algorithm has an access to "noisy" information about correlation between ϕ and y. (τ is the noise level. Noise can be adversarial)

$$|\tilde{q} - \mathbb{E}_{x,y}[\phi(x)y]| \le \tau \sim n^{-1/2}$$

- **Statistical query (SQ):** General query

$$|\tilde{q} - \mathbb{E}_{x,y}[\phi(x,y)]| \le \tau \sim n^{-1/2} \quad \text{(with } |\phi| \le 1\text{)}$$

➢ Analogy: $\tau \sim n^{-1/2}$ as the i.i.d. concentration error.

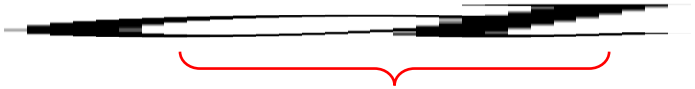
Theorem (informal [Damian et al. 22; Abbe et al. 23; Damian et al. 24])

To learn polynomial f_* with information exponent k (using **polynomial compute**):

- **CSQ learner** requires $\boldsymbol{n \gtrsim d^{k/2}}$ samples.
- **SQ learner** requires $\boldsymbol{n \gtrsim d^{\mathrm{ge}/2}}$ samples.

(we used the i.i.d. analogy)

Difficulty of learning single index model



Correlation query

SQ lower bound

$\tilde{O}(d^{(\mathrm{ge}/2)\mathrm{v}1})$

CSQ lower bound

d $\tilde{O}(d^{(\mathrm{ge}-1)\mathrm{v}1})$ $\tilde{O}(d^{k/2})$ $\tilde{O}(d^{k-1})$ $O(d^p)$

Information theoretic limit **MFLD**

SQ algorithm [Chen&Meka, 20]

Smoothed SGD [Damian et al. 23]

One pass SGD [Ben-Arous et al. 22]

Kernel method [Ghorbani et al. 21]

Gradient algorithm: CSQ

Complexity of non-feature learning methods

[Dandi et al. 2024][Lee, Oko, Suzuki, Wu; NeurIPS2024]

Gradient method with batch-reuse can implement SQ algorithm so that we obtain $\underline{n = \tilde{O}(d)}$ sample complexity. [**Generative exponent**]

What happens for in-context learning?

Generative exponent of polynomial

Theorem [Oko et al. 2024; Mondelli & Montanari 18; Barbier et al. 19; Chen & Meka 20]

For any polynomial σ_*,

$$\mathrm{ge}(\sigma_*) = \begin{cases} 1 & (\sigma_* \text{ is not even}) \\ 2 & (\sigma_* \text{ is even}) \end{cases}$$

Moreover, the generative exponent is achieved by ${\sigma^*}^j$ for some j:

$$\exists j \text{ s.t. } \mathrm{ge}(\sigma_*) = \mathrm{IE}({\sigma_*}^j)$$

Basic idea

Applying exponential function lower the information exponent upto the generative exponent:

$$\mathrm{ge}(\sigma_*) = \mathrm{IE}(\overline{\exp}(\sigma_*))$$
$$\mathrm{ge}(\sigma_*) = \mathrm{IE}(\sigma_* \cdot \overline{\exp}(\sigma_*))$$

⇒ Softmax can lower the information exponent.

Linear attention:

$$\frac{1}{n}\sum_{i=1}^{n} y_i \cdot x_i^\top \Gamma x_{\mathrm{qr}} \qquad \text{(CSQ)}$$

Softmax attention:

$$\frac{1}{C}\sum_{i=1}^{n} y_i \exp(y_i) \cdot \exp\left(x_i^\top \Gamma x_{\mathrm{qr}}\right) \qquad \text{(SQ)}$$

Nonlinear feature extraction

Proposition

If (a) Γ^* is the projection to $\mathcal{S}$ and (b) $n \geq \Omega(r^{3\mathrm{ge}(\sigma_*)/2})$, then

$$\mathrm{Attn}(\mathbf{X}_t, \mathbf{y}_t, x_{\mathrm{qr}}) = \sum_{i=1}^{n} y_{i,t} \frac{\exp(y_{i,t}/\rho)\exp(x_{i,t}^\top \Gamma^* x_{\mathrm{qr},t}/\rho)}{\sum_{i'=1}^{n}\exp(y_{i',t}/\rho)\exp(x_{i',t}^\top \Gamma^* x_{\mathrm{qr},t}/\rho)}$$

$$\simeq C_1 + C_2\left(\frac{\langle x_{\mathrm{qr}}, \beta_t\rangle}{\sqrt{r}}\right)^{\mathrm{ge}(\sigma_*)}$$

Attention layer can perform **test time feature extraction!**

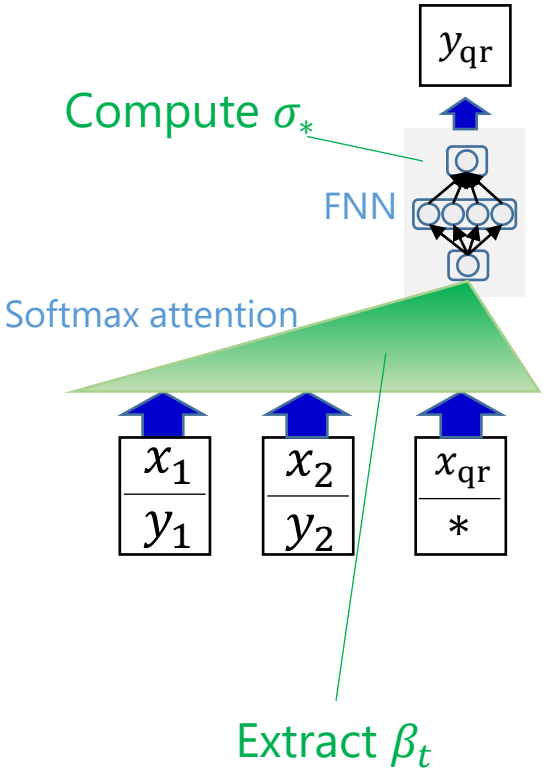
- **Test time feature learning:** Thanks to the nonlinear transformation $y \mapsto y\exp(y/\rho)$, the attention layer can perform feature extraction (i.e., can find the direction β_t from instructions) with smaller data size $r^{\frac{3\mathrm{ge}(\sigma_*)}{2}} (\ll r^{\mathrm{O}(k)} \ll r^{\mathrm{O}(P)})$. Linear attention
- **Pretraining feature learning:** The choice of Γ^* turns the sample complexity from $d^{\mathrm{O}(\mathrm{ge})}$ to $r^{\mathrm{O}(\mathrm{ge})}$.
  $\Rightarrow \Gamma^*$ should be found by the pretraining.

Transformer model

- **Softmax attention layer (single head):**

$$\mathrm{Attn}(\mathbf{X}_t, \mathbf{y}_t, x_{\mathrm{qr}}) = \sum_{i=1}^{n} y_{i,t} \frac{\exp(y_{i,t}/\rho)\exp(x_{i,t}^\top \Gamma x_{\mathrm{qr},t}/\rho)}{\sum_{i'=1}^{n} \exp(y_{i',t}/\rho)\exp(x_{i',t}^\top \Gamma x_{\mathrm{qr},t}/\rho)}$$

Nonlinear transformation
⇒ Out of CSQ framework



- **FNN layer:**

$$y_{\mathrm{qr}} = f_{\mathrm{TF}}(\mathbf{X}_t, \mathbf{y}_t, x_{\mathrm{qr}}; \Gamma, v, b, a) = \sum_{j=1}^{m} a_j \sigma\left(v_j \mathrm{Attn}(\mathbf{X}_t, \mathbf{y}_t, x_{\mathrm{qr}}) + b_j\right)$$

(σ: ReLU)

- The nonlinear transformation $y \mapsto y\exp(y/\rho)$ lower its information exponent. (actually, to generative exponent).
- The attention layer extracts $\langle x_{\mathrm{qr}}, \beta_t \rangle$.

Optimization algorithm

$$f_{\mathrm{TF}}(\mathbf{X}_t, \mathbf{y}_t, x_{\mathrm{qr}}; \Gamma, v, b, a) = \sum_{j=1}^{m} a_j \sigma\left(v_j \mathrm{Attn}_\Gamma(\mathbf{X}_t, \mathbf{y}_t, x_{\mathrm{qr}}) + b_j\right)$$

$$\mathrm{Attn}_\Gamma(\mathbf{X}_t, \mathbf{y}_t, x_{\mathrm{qr}}) = \sum_{i=1}^{n} y_{i,t} \frac{\exp(y_{i,t}/\rho)\exp(x_{i,t}^\top \Gamma x_{\mathrm{qr},t}/\rho)}{\sum_{i'=1}^{n}\exp(y_{i',t}/\rho)\exp(x_{i',t}^\top \Gamma x_{\mathrm{qr},t}/\rho)}$$

Initialize $\boldsymbol{w}_j^{(0)} \sim \mathrm{Unif}(\mathbb{S}^{d-1})$, $b_j = 0$, $\Gamma_{j,j}^{(0)} = \mathrm{Unif}(\{\pm 1\})$ (diagonal).

- **Stage 1: Optimization of Γ (attention).**

  Perform one step gradient descent w.r.t. Γ*

Train the attention to extract the coefficient β_t.

$$\Gamma^* = \Gamma^{(0)} - \eta_1 \nabla_\Gamma \left\{ \frac{1}{T_1} \sum_{t=1}^{T_1} \left(y_{\mathrm{qr},t} - f(X_t, Y_t, x_{\mathrm{qr},t}; \Gamma^{(0)}, v^{(0)}, b^{(0)}, a^{(0)}) \right)^2 + \lambda \|\Gamma\|_F^2 \right\}$$

- Analogous to one-step GD for 2-layer NN [Damian et al. 22; Ba et al. 22].

- **Stage 2: Optimization of a (FNN).**

  We fix v and b, and optimize a by gradient descent

Estimate σ^*

$$a^* \leftarrow \arg\min_{a \in \mathbb{R}^m} \frac{1}{T_2} \sum_{t=T_1+1}^{T_1+T_2} \left(y_{\mathrm{qr},t} - f(X_t, Y_t, x_{\mathrm{qr},t}; \Gamma^*, v^{(0)}, b^{(0)}, a) \right)^2 + \lambda \|a\|^2$$

Main result

Theorem (ICL risk bound)

1. Optimization guarantee

If $T_1 = \widetilde{\Omega}(r^2 d^{\mathrm{IE}(\sigma^*)+2})$, $n = \widetilde{\Omega}\,(r^2 d^{\mathrm{IE}(\sigma^*)+2})$, $T_2 = \widetilde{\Omega}(r^{3\mathrm{ge}(\sigma^*)/2})$, then the training error is $o_d(1)$.

2. Inference time sample complexity

If $N_{\mathrm{test}} = \widetilde{\Omega}\,(r^{3\mathrm{ge}(\sigma^*)/2})$, then the test error is $o_d(1)$.

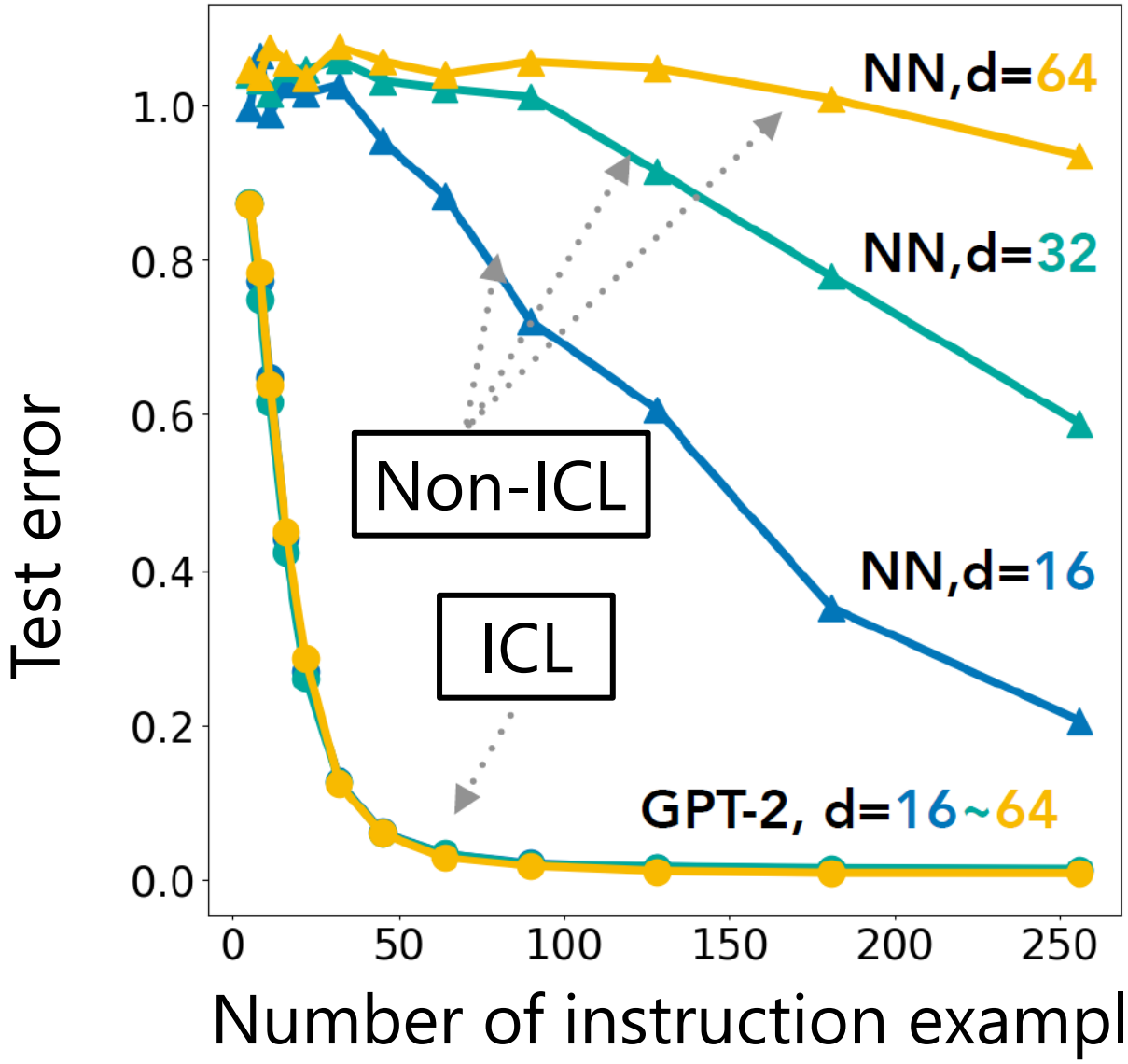
T_1: number of tasks in Stage 1 (learning Γ), T_2: number of tasks in Stage 2 (learning v, a, b), n: number of examples in each pretraining-tas

Benefit of nonlinear attention

	w/o pretraining		w/ pretraining	
Method	Kernel	NN	ICL/linear attention	ICL/**nonlinear attention**
Test time complexity	d^P	$d^{\mathrm{ge}(\sigma^*)/2}$	r^{4P} [Oko, Song, S, Wu, NeurIPS2024]	$r^{3\mathrm{ge}(\sigma^*)/2}$
Pretraining	---	---	$T_1 = d^{k+1}, n = d^k$	$T_1 = r^2 d^{k+2}, n = r^2 d^{k+2}$

Experiments on GPT-2

r is fixed. Only d is varied.

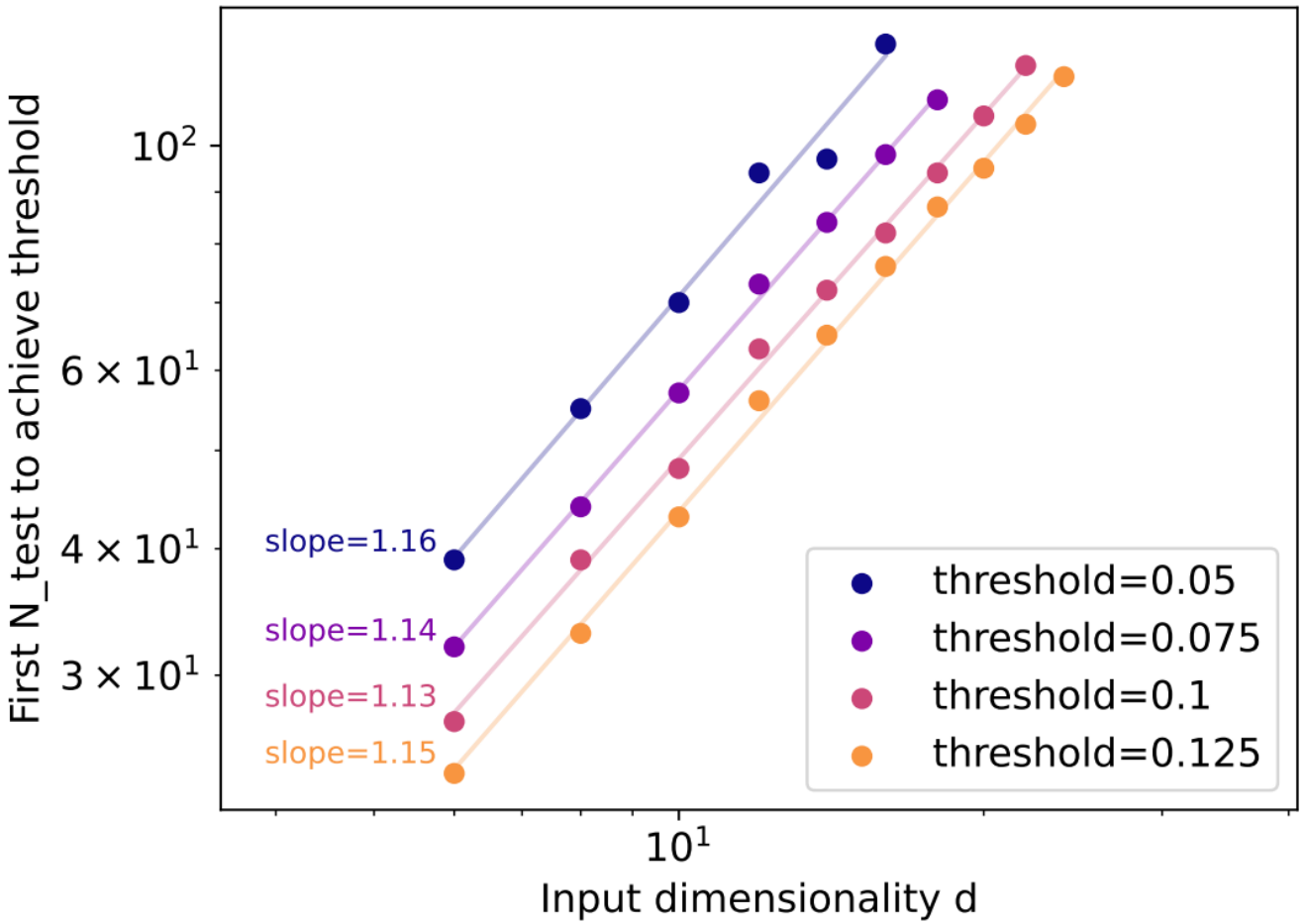


NN without pre-training is affected by d.

ICL by transformer is not affected by d.
(thanks to feature learning in pre-training)

Checking the exponent

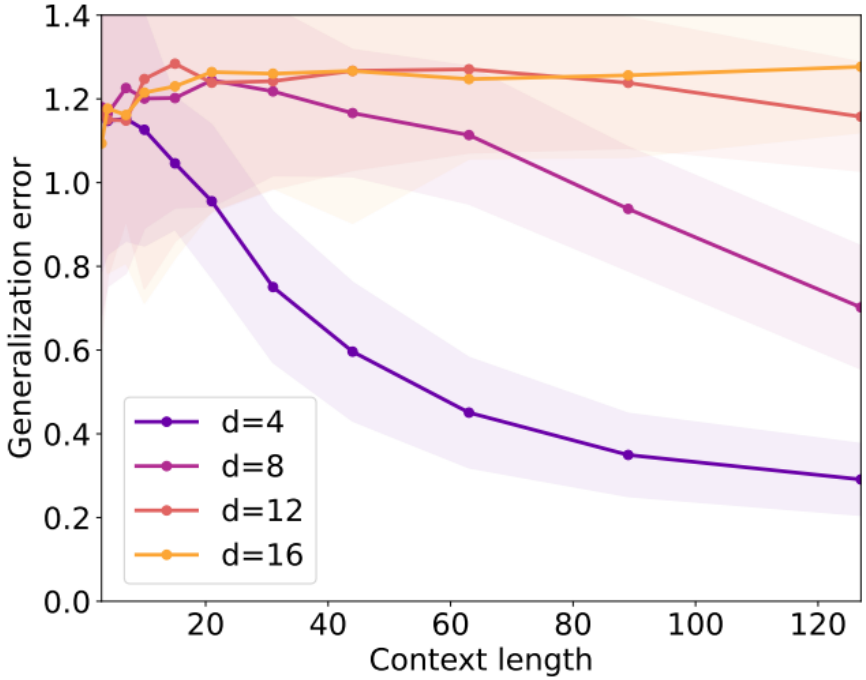
$d = r$ $\sigma_*(\cdot) = \mathrm{He}_3(\cdot)$ GPT-2 (6layers)



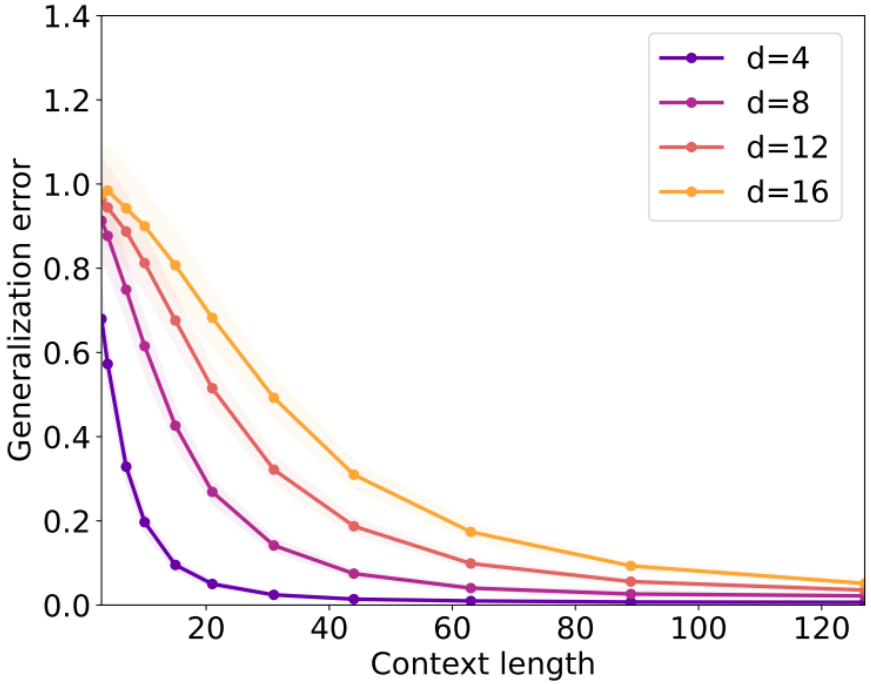
$n \simeq d^{1.1}$: Better than the CSQ lower bound $n = \Omega(d^{1.5})$.

Numerical experiments

Experiments for $d = r$.



Kernel ridge regression
Sensitive to d
⇒ Maximum degree



GPT-2 (6layers)
ICL
Less sensitive to d
⇒ Generative exponent

Follow-up work

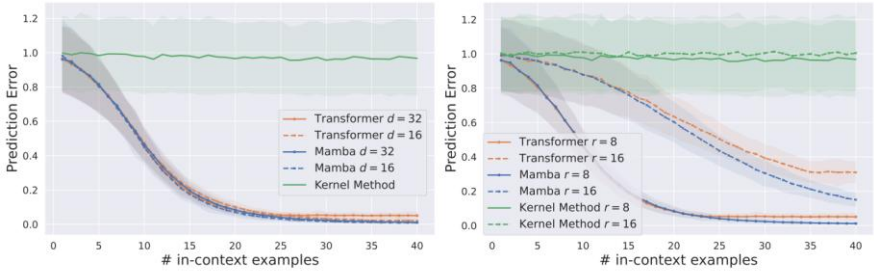
- **Mamba** [Gu and Dao, 2024]

[Junsoo Oh, Wei Huang, Taiji Suzuki: Mamba Can Learn Low-Dimensional Targets In-Context via Test-Time Feature Learning. arXiv:2510.12026.]

The nonlinear gate of Mamba also lower the information exponent:

If $N_{\text{test}} \geq \Omega(r^{3\text{ge}})$

$$\text{Mamba}(\boldsymbol{Z}; \gamma^*) \approx P_1 + P_2 \left(\frac{\langle \boldsymbol{\beta}, \boldsymbol{x} \rangle}{r} \right)^{\text{ge}(g_*)}$$



(a) $d = 32$ vs. $d = 16$ *(b)* $r = 8$ vs. $r = 16$

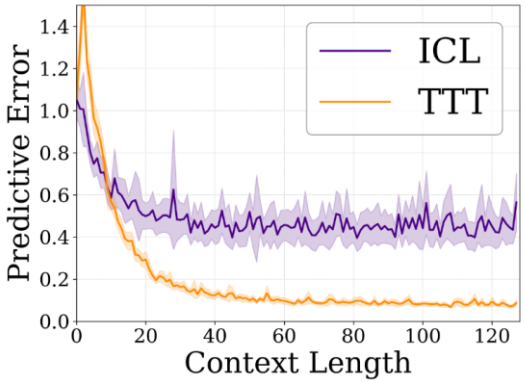
- **Test time training**

[Kento Kuwataka, Taiji Suzuki: Test time training enhances in-context learning of nonlinear functions. arXiv:2509.25741]

[Hu et al., 2025; Bertrand et al., 2023; Akyurek et al., 2025]

- Explicit convergence rate in test time:

$$\text{If } N_{\text{test}} = \widetilde{\Omega}\left(r^{\frac{3\text{ge}(\sigma^*_{\text{test}})}{2}}\right), \quad (\text{test loss}) \lesssim \sqrt{\frac{r^{3/2}}{N_{\text{test}}}}$$



- Link function can be varied for each task t:

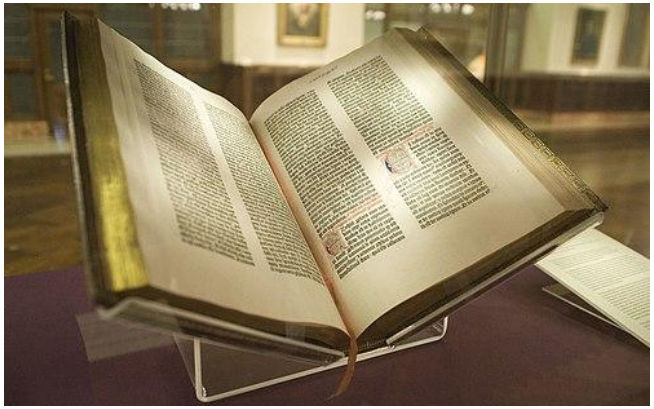
$$f_*^t(x) = \sigma_*^t(\langle x, \beta_t \rangle)$$

Transformer with infinite context length

[Kawata Ryotaro, Taiji Suzuki: Transformers as measure-theoretic associative memory: A statistical perspective and minimax optimality. ICLR2026]

Infinite length context

Long (infinite-length) context



[Wikipedia]

Answer

$F^*(\mu_0, x_{\mathrm{qr}})$

Token distribution

Embedding space

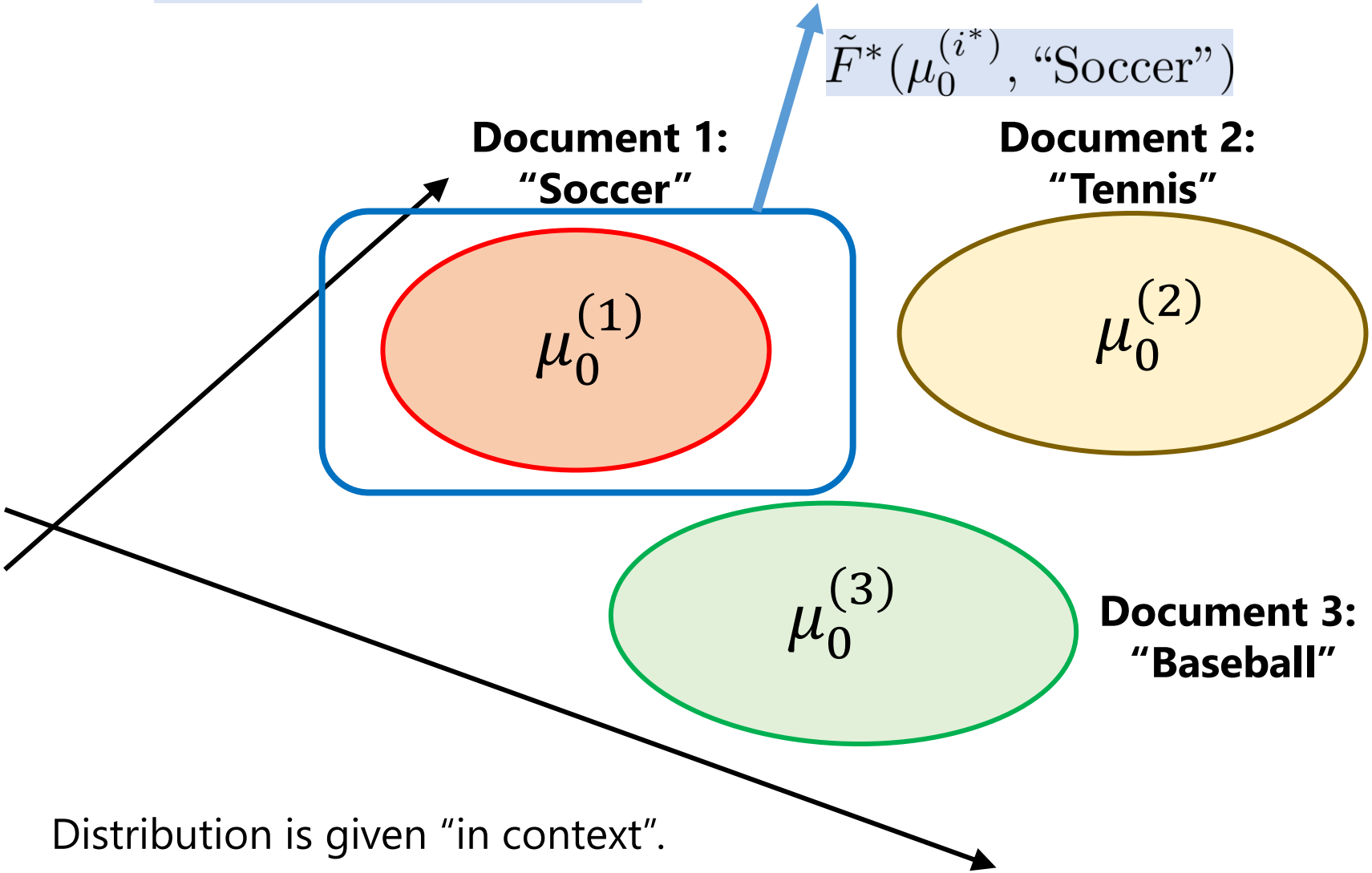
Mixture of topics

Query: "Soccer"

Answer

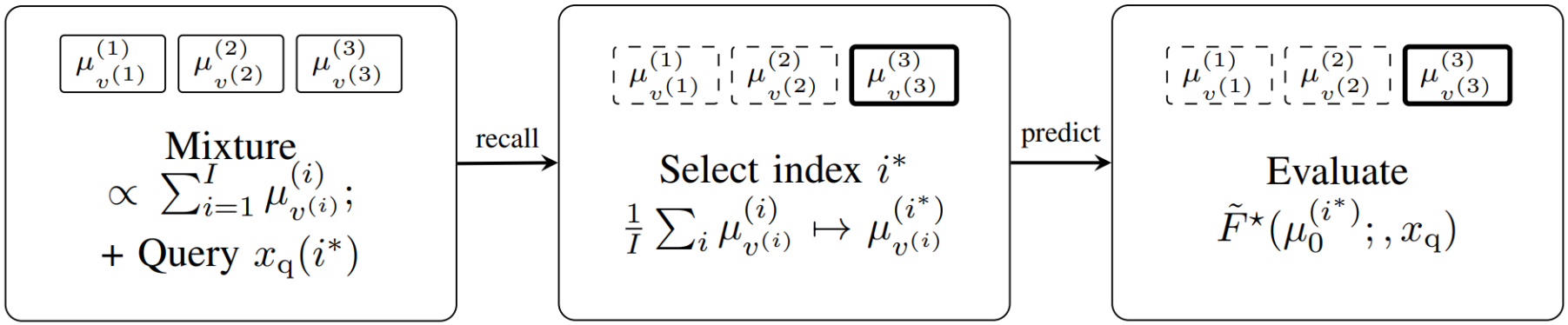
Regression from $\mathcal{P}$ to $\mathbb{R}$.

$\tilde{F}^*(\mu_0^{(i^*)}, \text{"Soccer"})$



Distribution is given "in context".

Problem setting



Associative recall at the level of measures.

1. Mixture documents and queries:

$$\mu^{(i)}_{v^{(i)}} := \left(v^{(i)}, \mu_0^{(i)}\right) \in \mathbb{S}^{d_0-1} \times \mathcal{P}(\mathcal{X}_0)$$

Document label — Distribution of contents (e.g., word embeddings)

Prompt input: $\nu = \frac{1}{I}\sum_{i=1}^{I} \mu^{(i)}_{v^{(i)}}$ **Mixture of documents**

Query: $x_q = v^{(i^*)}$ for some $i^* \in [I]$.

We may easily extend it to $x_q = (v^{(i^*)}, z_q)$, but we stay the simplest setting.

2. Ground-truth recall-and-predict map:

$$y = F^*(\nu, x_{\mathrm{q}}) = \tilde{F}^*(\mu_0^{(i^*)}, x_{\mathrm{q}})$$

F^* must first *associate* the query with the relevant component i^* and then *predict* a scalar from the recalled context.

Transformer model

1. Attention layer (for measure input)

$$\mathrm{Attn}_\theta(\nu, x) = Ax + \sum_{h=1}^{H} W^h \int \mathrm{Softmax}(\langle Q^h x, K^h y\rangle)\, V^h y \,\mathrm{d}\nu(y)$$

$$\mathcal{A}(d_{\mathrm{attn}}, H, B_a, B_a', S_a, S_a') := \{\mathrm{Attn}_\theta \mid \max_h \{\|W^h\|_\infty, \|Q^h\|_\infty, \|K^h\|_\infty, \|V^h\|_\infty\} \le B_a,$$
$$\max_h \{\|W^h\|_0, \|Q^h\|_0, \|K^h\|_0, \|V^h\|_0\} \le S_a, \|A\|_\infty \le B_a', \ \|A\|_0 \le S_a', \ \|\mathrm{Attn}_\theta\|_\infty \le F\}$$

2. MLP layer

$$f : \mathbb{R}^{p_0} \to \mathbb{R}^{p_{\ell+1}}, \quad \mathbf{x} \mapsto f(\mathbf{x}) = W_\ell \sigma_{\mathbf{v}_\ell} W_{\ell-1} \sigma_{\mathbf{v}_{\ell-1}} \cdots W_1 \sigma_{\mathbf{v}_1} W_0 \mathbf{x}$$

where $\sigma_{\mathbf{v_i}}(\mathbf{z}) := \sigma(\mathbf{z} - \mathbf{v_i})$ with ReLU activation σ.

$$\mathcal{F}(\ell, \boldsymbol{p}, s, F) := \{f \text{ of the form above} \mid$$
$$\max_j\{\|W_j\|_\infty, |v_j|_\infty\} \le 1, \textstyle\sum_j \|W_j\|_0 + |v_j|_0 \le s, \ \|f\|_\infty \le F\}$$

3. Multilayer Transformer

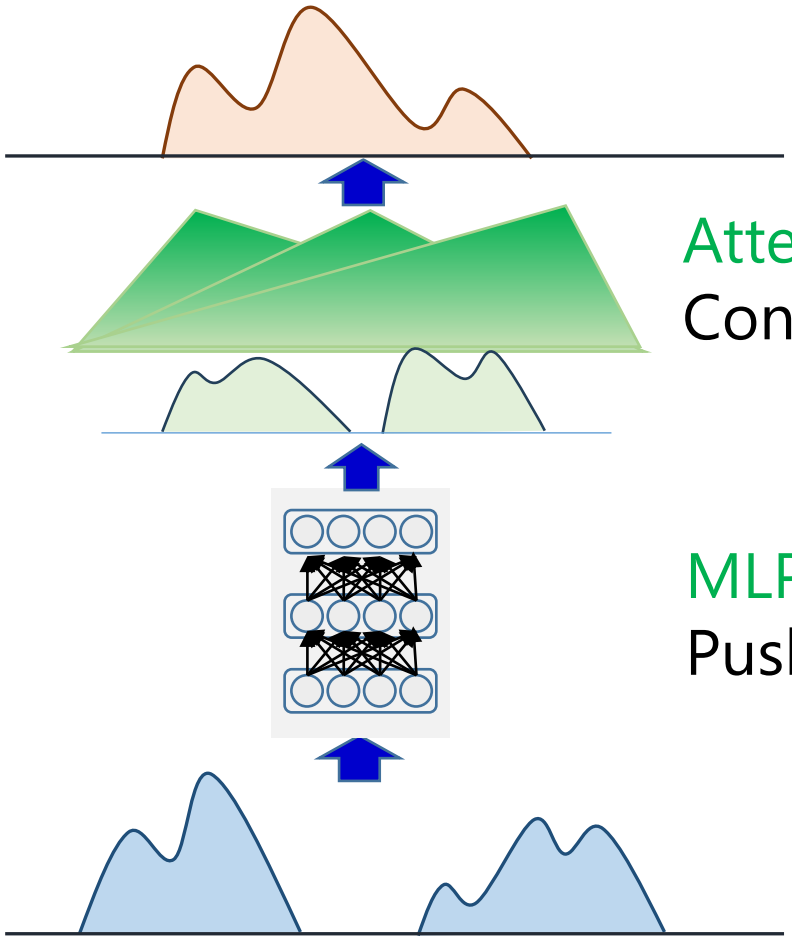
$$\mathrm{Attn}_{\theta_L} \diamond \mathrm{MLP}_{\xi_L} \diamond \cdots \diamond \mathrm{Attn}_{\theta_1} \diamond \mathrm{MLP}_{\xi_1}$$

where $(\Gamma_2 \diamond \Gamma_1)(\nu, x) := \Gamma_2(\mu_1, \Gamma_1(\nu, x))$, where $\mu_1 := (\Gamma_1(\nu, \cdot))_\sharp \nu$

[Furuya et al. (2025)]

As for MLP, we define $\mathrm{MLP}_{\xi_1}(\mu, x) := \mathrm{MLP}_{\xi_1}(x)$

Transformer as a measure map



Attention:
Convolution

MLP-layer:
Push-forward

Attention is a convolution operator:

$$\mathrm{Attn}_\theta(\nu, x) = Ax + \sum_{h=1}^{H} W^h \int \mathrm{Softmax}(\langle Q^h x, K^h y\rangle)\, V^h y \, \mathrm{d}\nu(y)$$

Assumption

- **Density is in an RKHS**
  - Suppose that $\mathcal{X}_0 \subset [-M, M]^{d_2}$ is a bounded domain (tokens are embedded in $\mathcal{X}_0$).

$\mu_0^{(i)}$ has a density $p_{\mu_0^{(i)}}$ which can be expanded as (e_j: ONS in L^2)

$$p_{\mu_0^{(i)}}(x) = \sum_{j=1}^{\infty} \alpha_j e_j(x) \quad \text{where} \quad \sum_{j=1}^{\infty} \lambda_j^{-1} \alpha_j^2 \leq 1$$

- Eigenvalue decay: $\lambda_j \asymp \exp(-cj^{\alpha})$
- Analytic eigenfunctions: $e_j(x) = \sum_{\boldsymbol{k} \in \mathbb{N}^{d_2}} a_{\boldsymbol{k}} x^{\boldsymbol{k}}$ (absolutely convergent on $[-M, M]^{d_2}$)

Included in the unit ball of RKHS $\mathcal{H}_0$ with the kernel $K(x, x') = \sum_{j \geq 1} \lambda_j e_j(x) e_j(x')$.
Gaussian kernel satisfies the assumption.

$$p_{\mu_0^{(i)}} \in B(\mathcal{H}_0)$$

- **Lipschitz continuity:**

$$|\tilde{F}^*(\mu_0, x_{\mathrm{q}}) - \tilde{F}^*(\mu_0', x_{\mathrm{q}}')| \lesssim \|\mu_0 - \mu_0'\|_{\mathcal{H}_0} + \|x_{\mathrm{q}} - x_{\mathrm{q}}'\|$$

e_j is unknown ⇒ Should be estimated by pre-training. [Feature learning]

Main result

Observation: $y_t = F^*(\nu_t, x_{\mathrm{q}_t}) + \xi_t$ where $\xi_t \sim N(0, \sigma^2)$.

$$\mathcal{S}_n := \{(\nu_t, x_{\mathrm{q}_t}, y_t)\}_{t=1}^n$$

$$\hat{F}_n \in \arg\min_{F \in \mathcal{F}} \frac{1}{n} \sum_{t=1}^{n} \left(y_t - F(\nu_t, x_{\mathrm{q}_t})\right)^2$$

$\mathcal{F}$: 2-layer Transformer

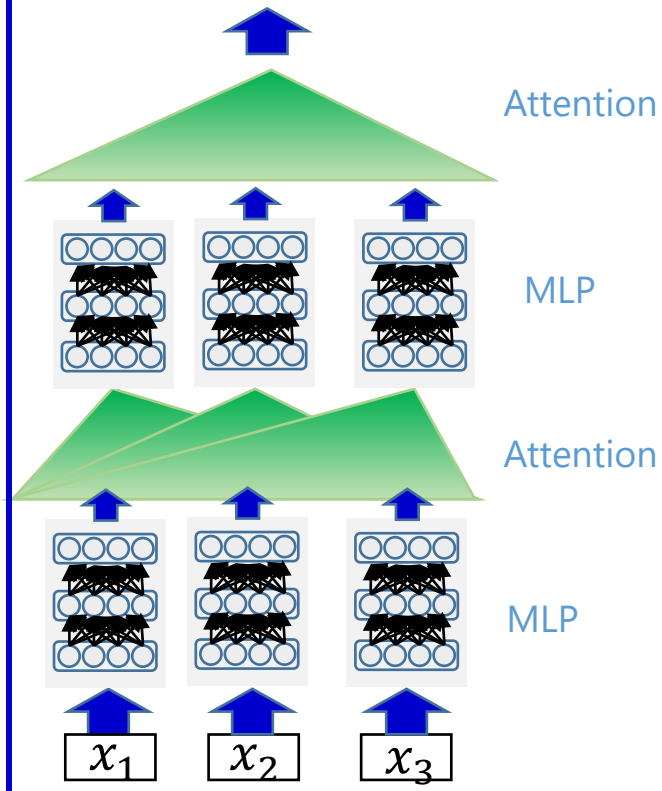
$$R(F^\star, \hat{F}) := \mathbb{E}_{\mathcal{S}_n}\left[\left\|F^\star(\nu, x_{\mathrm{q}}) - \hat{F}(\nu, x_{\mathrm{q}})\right\|^2_{L^2(\mathbb{P}_{\nu, x_{\mathrm{q}}})}\right].$$

Theorem (Generalization error)

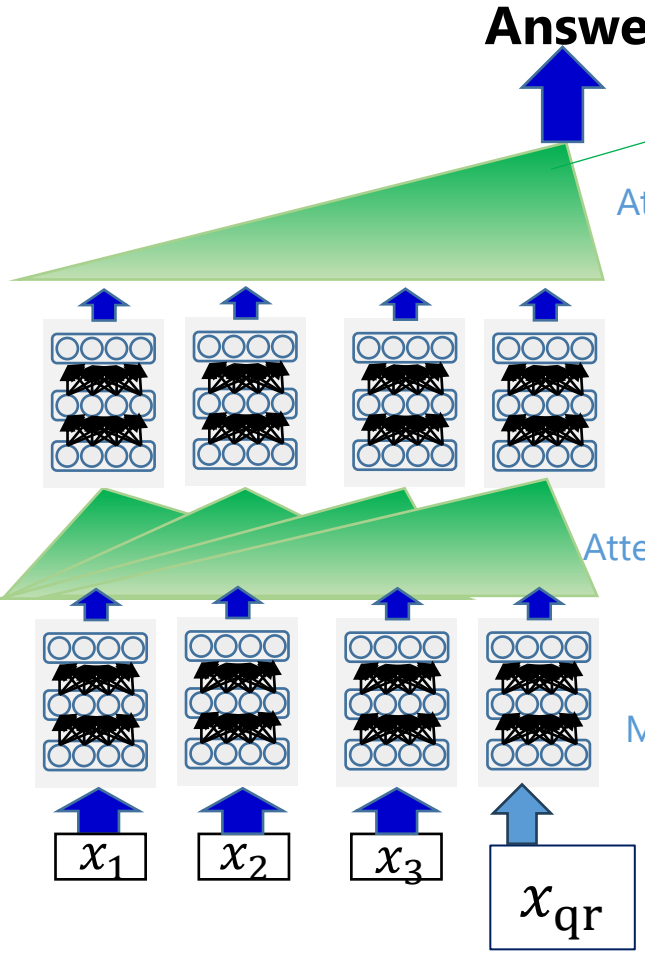
Suppose that the number of mixture components satisfies $I \leq d_1 \leq O((\log n)^{\frac{1}{\alpha+1}})$.
For an appropriately chosen architecture parameters, $\hat{F}$ achieves

$$R(F^\star, \hat{F}_n) \lesssim \exp\left\{-\Omega\left((\log n)^{\alpha/(\alpha+1)}\right)\right\}$$

2-layer Transformer



Proof idea



Extract the answer

$\tilde{F}^*(\mu_0^{(i^*)}, x_{\mathrm{qr}})$

Compute the Lipschitz continuous function from $\mathcal{H}_0$ to $\mathbb{R}$.

Extract the coefficients α_j of the density $\mu_0^{(i^*)}$ (embedding to RKHS $\mathcal{H}_0$).

Compute basis function $\left(e_j(x)\right)_{j=1}^{N}$.

$\left(\int e_j(x)\mathrm{d}\mu_0^{(i^*)}\right)_{j=1}^{N}$

- Should identify appropriate basis functions through pre-training
- Obtain an appropriate representation of the distribution.

- N
- $\log(V(\epsilon)) \lesssim \epsilon^{-\frac{\alpha}{\alpha+1}}$ (Covering num)
- $\epsilon \simeq \exp(-c(\ln n)^{\alpha/(1+\alpha)})$

$R \lesssim \epsilon + \frac{\cdot}{n}$

Minimax optimality

Theorem (Minimax optimal rate)

Under the same setting,

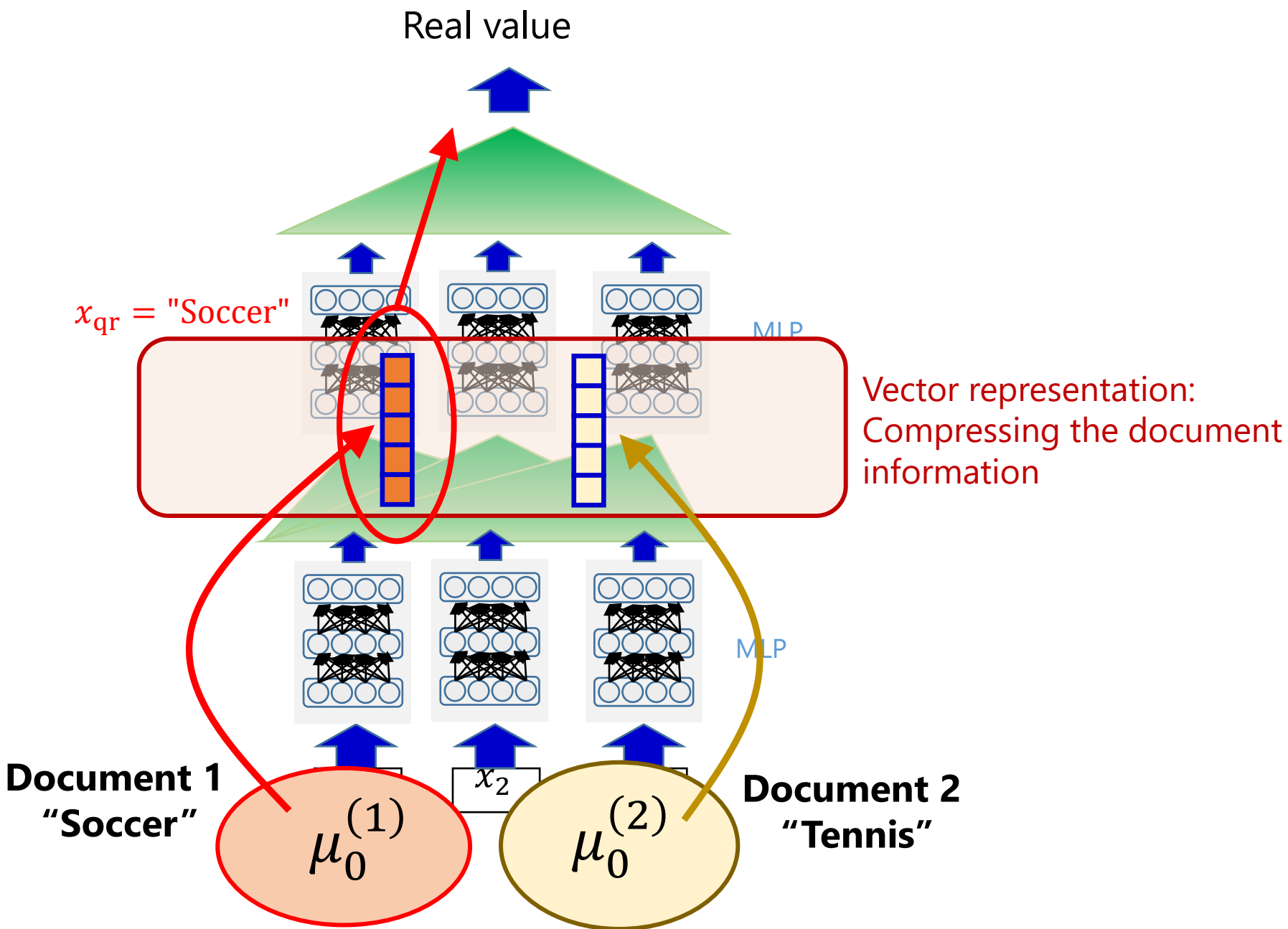
$$\inf_{\hat{\mathcal{F}}_n} \sup_{F^* \in \mathcal{F}^*} R(F^\star, \hat{F}_n) \gtrsim \exp\left\{ -\Omega\left((\log n)^{\alpha/(\alpha+1)}\right)\right\}$$

- Transformer achieves the minimax optimal rate.
- **Key lemma:** Packing number of Lipschitz functions

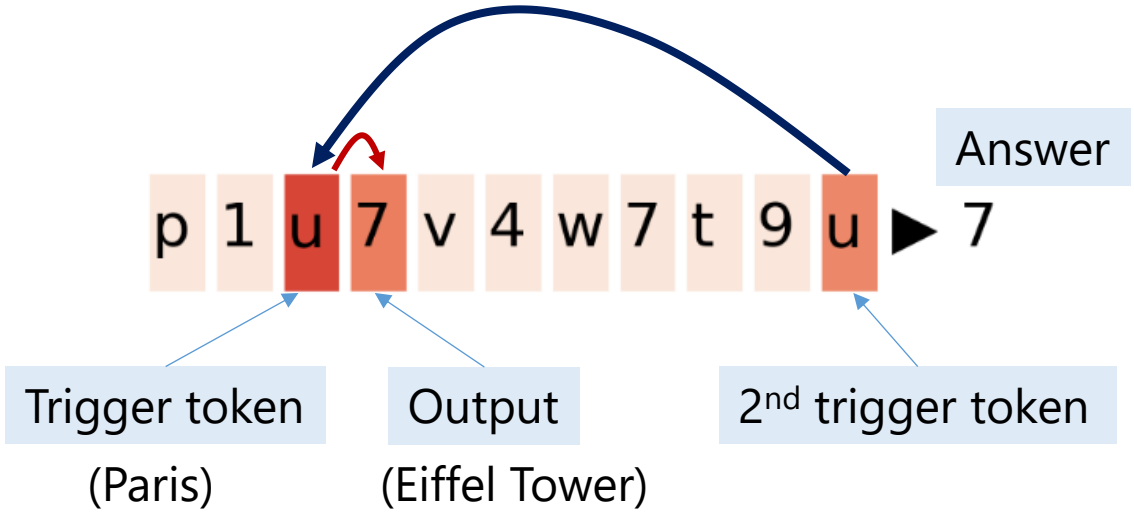
Lemma (Lanthaler (2024))

$$\log M(\mathrm{Lip}_1([0,1]^d, \|\cdot\|_\infty); \epsilon; L^p([0,1]^d)) \gtrsim \left(\frac{c}{d\epsilon}\right)^d, \quad \forall \epsilon \in (0, c/d].$$

Set of 1-Lipschitz continuous functions w.r.t. L^∞-norm


Real value
x_{qr} = "Soccer"
MLP
Vector representation:
Compressing the document information
MLP
Document 1
"Soccer"
x_2
$\mu_0^{(1)}$
$\mu_0^{(2)}$
Document 2
"Tennis"

Related work: Induction head



[Elhage et al. 2021, Olsson et al. 2022, Bietti et al. 2023]

- **Two scenarios:**
  - **Short cut**: The model just memorizes the position of trigger token
  - **Induction head**: They learn how to find the trigger tok
    ⇒ It can generalize for OOD test data **[feature learnin**

$$R(\mathcal{D}_\ell) = \frac{\sum_{\ell \in \mathcal{S}} \ell^{-1} q_\ell}{\max_{\ell \in \mathcal{S}} \ell^{-1} q_\ell}$$

Theorem [Kawata, Song, Bietti, et al. NeurIPS2025]

There exists $\epsilon_1, \epsilon_2 = \Theta(N_{\mathrm{trg}})$ such that

1. $R(\mathcal{D}_\ell) > \epsilon_1$ **(high diversity):** The pretrained transformer generalizes OOD.
2. $R(\mathcal{D}_\ell) < \epsilon_2$ **(low diversity):** There exist OOD test sequences such that the pretrained transformer fails.

Induction head
(Do not memorize position)

Memorize position

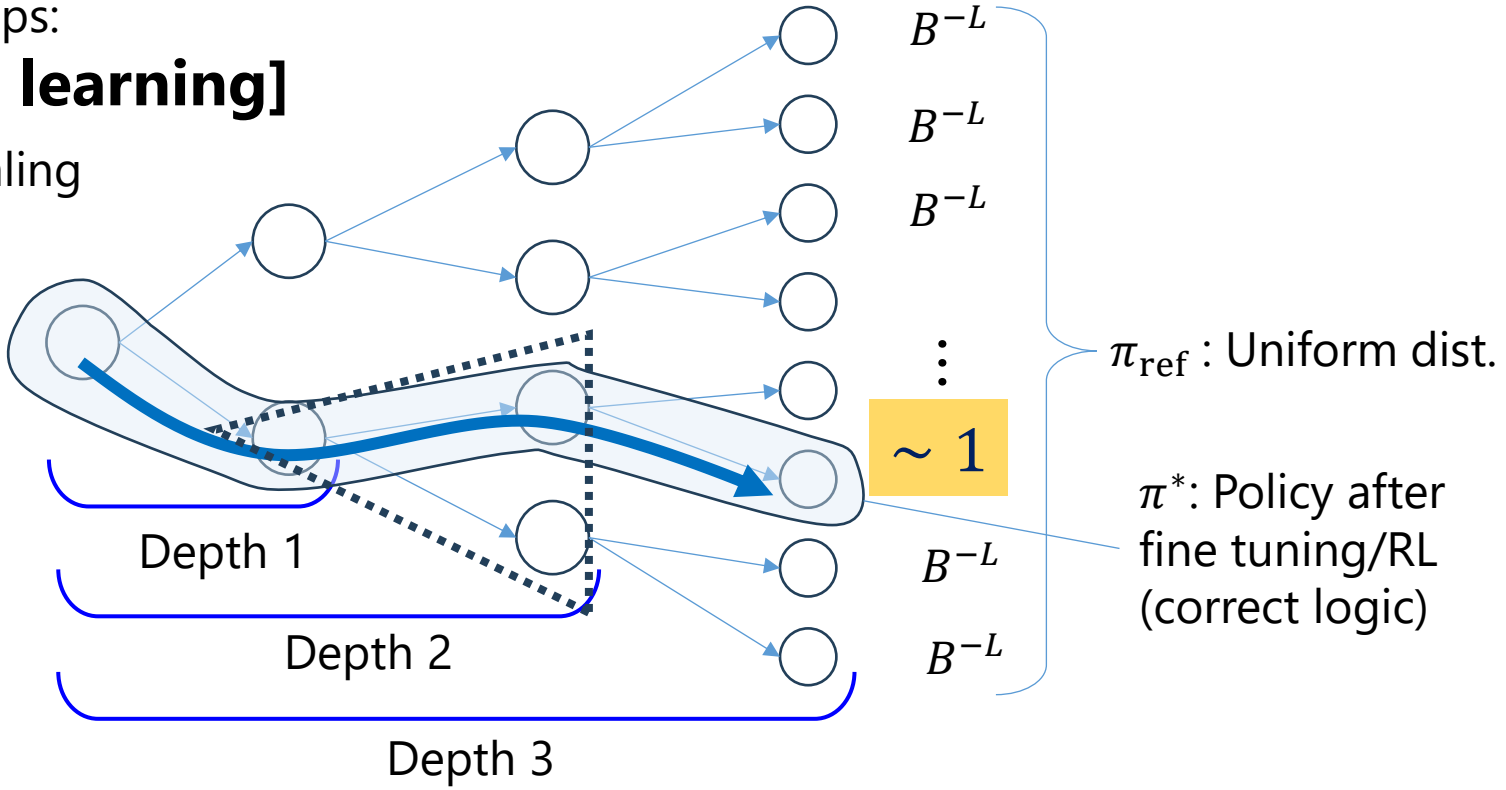
Other topics: Curriculum learning

[Dake, Huang, Han, Nitanda, Wong, Zhang, Suzuki: Provable Benefit of Curriculum in Transformer Tree-Reasoning Post-Training. ICML2026]

By training the model at each difficulty stage, we may decompose the difficulty into much smaller steps:

[Curriculum learning]

- Test time scaling
- Post training



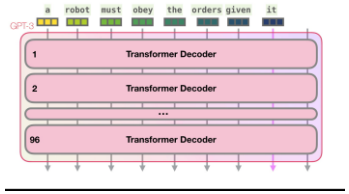
Required sample complexity [k-parity problem]:

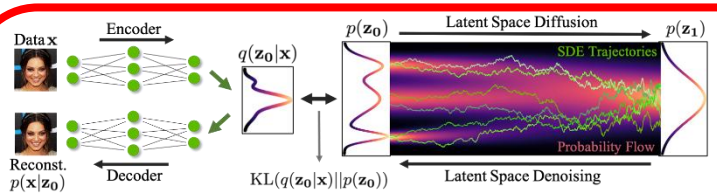
Thm **Direct training:** $d^{2(k+1)}$ **Curriculum learning:** $(k+1)d^2$ with optimization guarantee

The efficiency can be improved **exponential order**.

See also [Liu, Farina, Ozdaglar: UFT: Unifying Supervised and Reinforcement Fine-Tuning. arXiv:2505.16984, 2025]

Presentation overview





[Vahdat, Kreis, Kautz: Score-based Generative Modeling in Latent Space. arXiv:2106.05931]

Part 1: In-context learning

Test time feature learning by softmax attention

- Achieving the sample complexity with generative exponent
- Gaussian single index model

Measure theoretic in-context learning

- Infinite-length context
- Measure to real-value problem

Part 2: Diffusion models

Minimax optimality

- Adaptation to smoothness
- Adaptation to low-dim subspace

General data support

- Curved manifold
- Anisotropic smoothness

- [In-context learning] Naoki Nishikawa, Yujin Song, Kazusato Oko, Denny Wu, Taiji Suzuki: Nonlinear transformers can perform inference-time feature learning: a case study of in-context learning on single-index models. ICML2025.
- [Measure theoretic associative recall] Kawata Ryotaro, Taiji Suzuki: Transformers as measure-theoretic associative memory: A statistical perspective and minimax optimality. ICLR2026.
- [Diffusion models] (1) Kazusato Oko, Shunta Akiyama, Taiji Suzuki: Diffusion Models are Minimax Optimal Distribution Estimators. ICML2023. (2) Guoji Fu, Taiji Suzuki, Wee Sun Lee, Atsushi Nitanda: Beyond the OU Process: Dimension-Adaptive Drifts for Minimax OptimalScore-Based Generative Modeling on Low-dimensional Manifolds. 2026. (3) Yuto Maki, Taiji Suzuki: Efficiency of Diffusion Models for Infinite-dimensional Data Generation: Dimension Independent Approximation and Estimation Errors. 2026.

Diffusion model

「An astronaut riding a horse in a photorealistic style」

「Teddy bears shopping for groceries in the style of ukiyo-e」

DALL·E: [Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, Ilya Sutskever: Zero-Shot Text-to-Image Generation. ICML2021.]
DALL·E2:[Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, Mark Chen: Hierarchical Text-Conditional Image Generation with CLIP Latents. arXiv:2204.06125]

High dimension

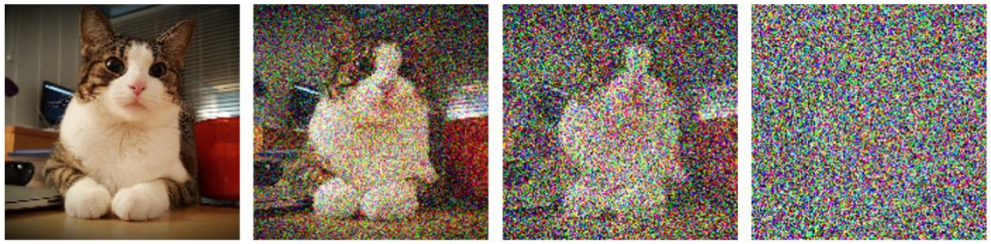
SORA
(OpenAI, 2024)

Diffusion model

[Sohl-Dickstein et al., 2015; Song & Ermon, 2019; Song et al., 2020; Ho et al., 2020; Vahdat et al., 2021]

Forward process：Convert the target distribution to a noise distribution (e.g., Gaussian)

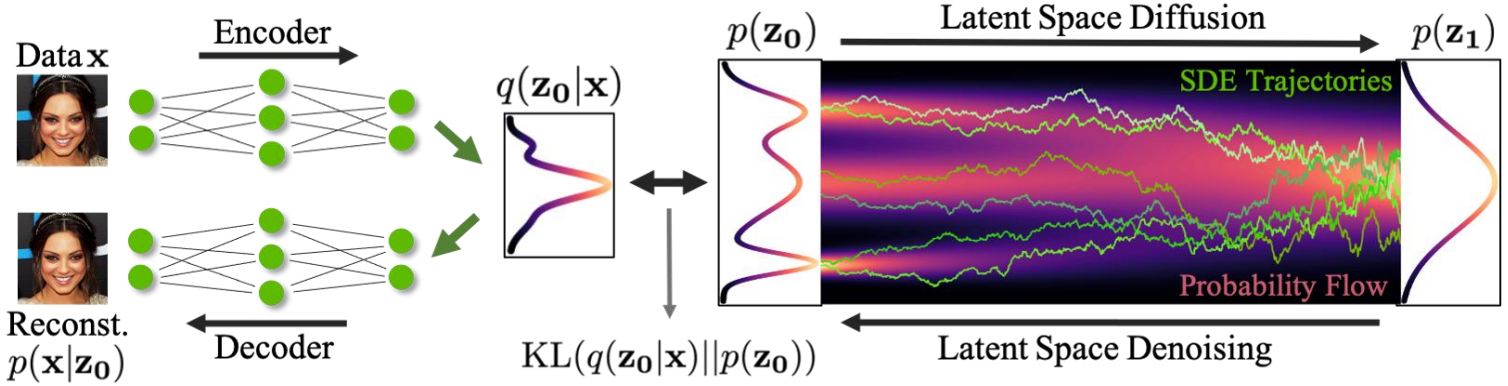
$$\mathrm{d}X_t = -X_t \mathrm{d}t + \sqrt{2}\mathrm{d}B_t$$



$$\mathrm{d}Y_t = (Y_t + 2\nabla \log(p_{\overline{T}-t}(Y_t)))\mathrm{d}t + \sqrt{2}\mathrm{d}B_t$$

$(Y_t \sim X_{\overline{T}-t})$

Reverse process：Convert the noise distribution to the target distribution



[Vahdat, Kreis, Kautz: Score-based Generative Modeling in Latent Space. arXiv:2106.05931]

Forward process

OU-process

Forward process: $\mathrm{d}X_t = -X_t \mathrm{d}t + \sqrt{2}\mathrm{d}B_t$

Wasserstein gradient flow to minimize the KL-divergence from $N(0, I)$.

$$\mathrm{KL}(p_t||N(0,I)) \leq \exp(-2t)\mathrm{KL}(p_0||N(0,I))$$

(Bakry-Emery criterion)

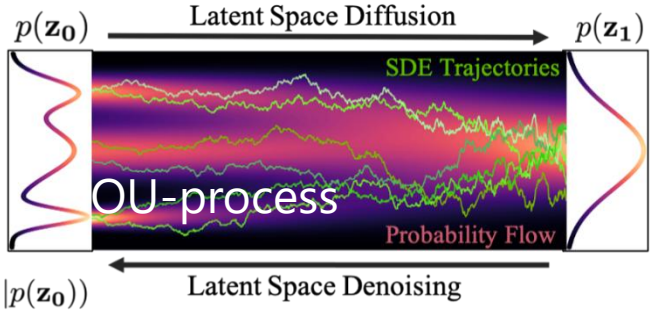
$$\mu^*(x) \leftarrow N(0,I)$$

$$\mathrm{d}X_t = \nabla \log(\mu^*(X_t))\mathrm{d}t + \sqrt{2}\mathrm{d}B_t$$

$$\begin{aligned}\partial_t p_t &= \nabla \cdot [p_t(-\nabla \log \mu^*)] + \Delta_x p_t \\ &= \nabla \cdot [p_t \nabla \log (p_t/\mu^*)] \\ &= \nabla \cdot \left[p_t \nabla \frac{\delta \mathrm{KL}(p_t||\mu^*)}{\delta p_t}\right]\end{aligned}$$

Fokker-Planck equation

minimizing $\mathbb{E}_p[-\log(\mu^*)] + \mathrm{Ent}(p)$



Standard normal

[Vahdat, Kreis, Kautz: Score-based Generative Modeling in Latent Space. arXiv:2106.05931]

Reverse process

Reverse process:

$$Y_0 \sim p_{\overline{T}}$$

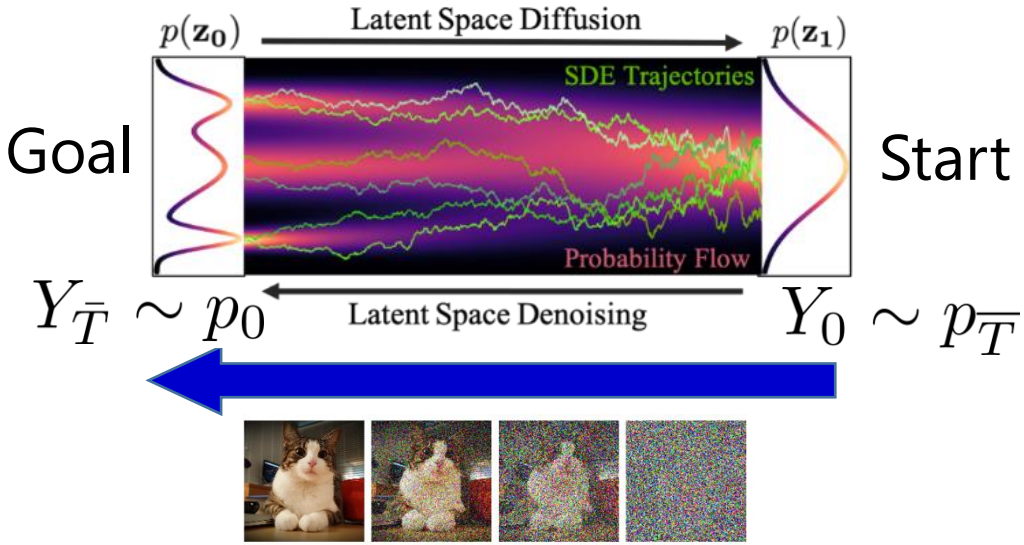
$$\mathrm{d}Y_t = (Y_t + 2\nabla \log(p_{\overline{T}-t}(Y_t)))\mathrm{d}t + \sqrt{2}\mathrm{d}B_t \quad (t \in [0, \overline{T}])$$

[Haussmann & Pardoux, 1986]

Fact：Y_t's distribution=$X_{\overline{T}-t}$'s distribution

That is, $Y_t \sim p_{\overline{T}-t}$. $\Rightarrow \boldsymbol{Y_{\overline{T}} \sim p_0}$

By following the forward process in reverse, noise that follows a (nearly) normal distribution can be gradually modified to reproduce the original distribution of the images.



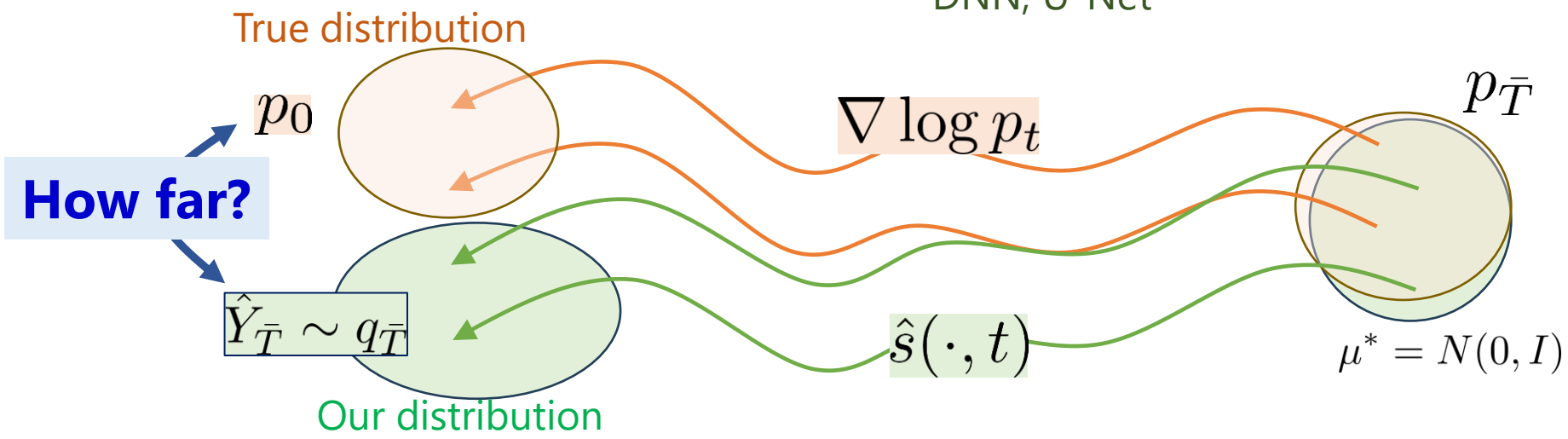
Approximation error bound

$$Y_0 \sim p_{\bar{T}} \qquad \mathrm{d}Y_t = (Y_t + 2\nabla \log(p_{\bar{T}-t}(Y_t)))\mathrm{d}t + \sqrt{2}\mathrm{d}B_t$$

Approximation ↓ Approximation

$$\hat{Y}_0 \sim N(0, I) \qquad \mathrm{d}\hat{Y}_t = (\hat{Y}_t + 2\hat{s}(\hat{Y}_t, \bar{T} - t))\mathrm{d}t + \sqrt{2}\mathrm{d}B_t$$

DNN, U-Net



$$\mathrm{KL}(q_{\bar{T}}||p_0) \leq \int_0^{\bar{T}} \mathbb{E}_{X_t}[\|\hat{s}(X_t, t) - \nabla \log(p_t(X_t))\|^2]\mathrm{d}t + \mathrm{KL}(p_{\bar{T}}||\mu^*)$$

KL-divergence — **Score estimation error** — **Forward process convergence** ($\leq \exp(-2\bar{T})$)

[Girsanov theorem]

[Chen, Lee, Lu: Improved Analysis of Score-based Generative Modeling: User-Friendly Bounds under Minimal Smoothness Assumptions. ICML2023]
[Chen, Chewi, Li, Li, Salim, and Zhang: Sampling is as easy as learning the score: theory for diffusion models with minimal data assumptions. ICLR2023]

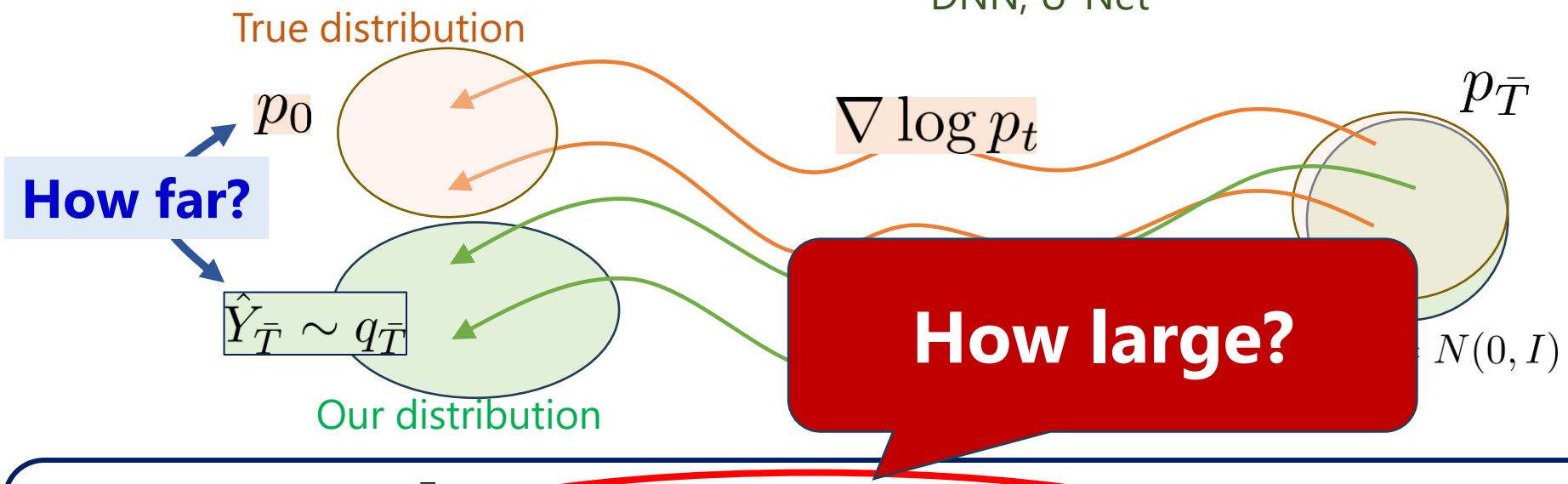
Approximation error bound

$$Y_0 \sim p_{\overline{T}} \qquad \mathrm{d}Y_t = (Y_t + 2\nabla \log(p_{\overline{T}-t}(Y_t)))\mathrm{d}t + \sqrt{2}\mathrm{d}B_t$$

Approximation ↓ Approximation

$$\hat{Y}_0 \sim N(0, I) \qquad \mathrm{d}\hat{Y}_t = (\hat{Y}_t + 2\hat{s}(\hat{Y}_t, \bar{T} - t))\mathrm{d}t + \sqrt{2}\mathrm{d}B_t$$

DNN, U-Net



$$\mathrm{KL}(q_{\bar{T}} || p_0) \leq \int_0^{\bar{T}} \mathbb{E}_{X_t}[\|\hat{s}(X_t, t) - \nabla \log(p_t(X_t))\|^2]\mathrm{d}t + \mathrm{KL}(p_{\bar{T}} || \mu^*)$$

KL-divergence

Score estimation error

Forward process convergence ($\leq \exp(-2\bar{T})$)

[Girsanov theorem]

[Chen, Lee, Lu: Improved Analysis of Score-based Generative Modeling: User-Friendly Bounds under Minimal Smoothness Assumptions. ICML2023]
[Chen, Chewi, Li, Li, Salim, and Zhang: Sampling is as easy as learning the score: theory for diffusion models with minimal data assumptions. ICLR2023]

Smoothness of the density

The true distribution p_0 is supported on $[-1,1]^d$ and

$$p_0 \in B^{\beta}_{p,q}$$

with $\beta > (1/p - 1/2)_+$ as a density function on $[-1,1]^d$.

$$p_0(x) \geq c \quad (\forall x \in [-1,1]^d)$$

Besov space ($B^{\beta}_{p,q}(\Omega)$)

Intuition

Smoothness

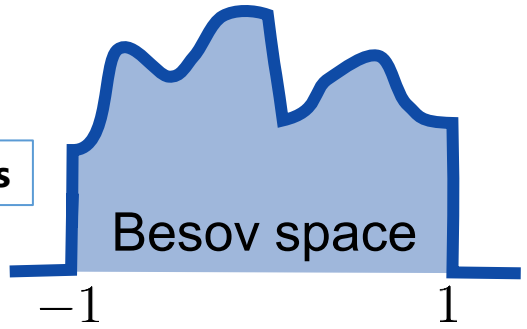
$$\|f\|_{B^{\beta}_{p,q}(\Omega)} = \|f\|_{L^p(\Omega)} + \|D^{\beta} f\|_{L^p(\Omega)}$$

Uniformity of smoothness

$$\|f\|_{B^{\beta}_{p,q}(\Omega)} = \|f\|_{L^p(\Omega)} + \left(\int_0^{\infty} [t^{-\beta} \omega_m(f,t)_p]^q \frac{\mathrm{d}t}{t} \right)^{1/q}.$$

Smoothness

Spatial inhomogeneity



Estimation error analysis

[Kazusato Oko, Shunta Akiyama, Taiji Suzuki: Diffusion Models are Minimax Optimal Distribution Estimators. ICML2023]

Stable diffusion, 2022.

$$\mathrm{d}X_t = -X_t \mathrm{d}t + \sqrt{2}\mathrm{d}B_t$$

Forward process

Backward process

$$\mathrm{d}Y_t = (Y_t + 2\nabla \log(p_{\overline{T}-t}(Y_t))\mathrm{d}t + \sqrt{2}\mathrm{d}B_t \quad (Y_t \sim X_{\overline{T}-t})$$

Empirical score matching estimator:

$$\hat{s} = \arg\min_{s \in \mathrm{DNN}} \frac{1}{n} \sum_{i=1}^{n} \int_{t=\underline{T}}^{\overline{T}} \mathbb{E}_{X_t|X_0=x_{0,i}} \left[\| s(X_t, t) - \nabla \log p_t(X_t|x_{0,i}) \|^2 \right] \mathrm{d}t$$

Theorem Let $\hat{Y}$ be the r.v. generated by the backward process w.r.t. $\hat{s}$, then

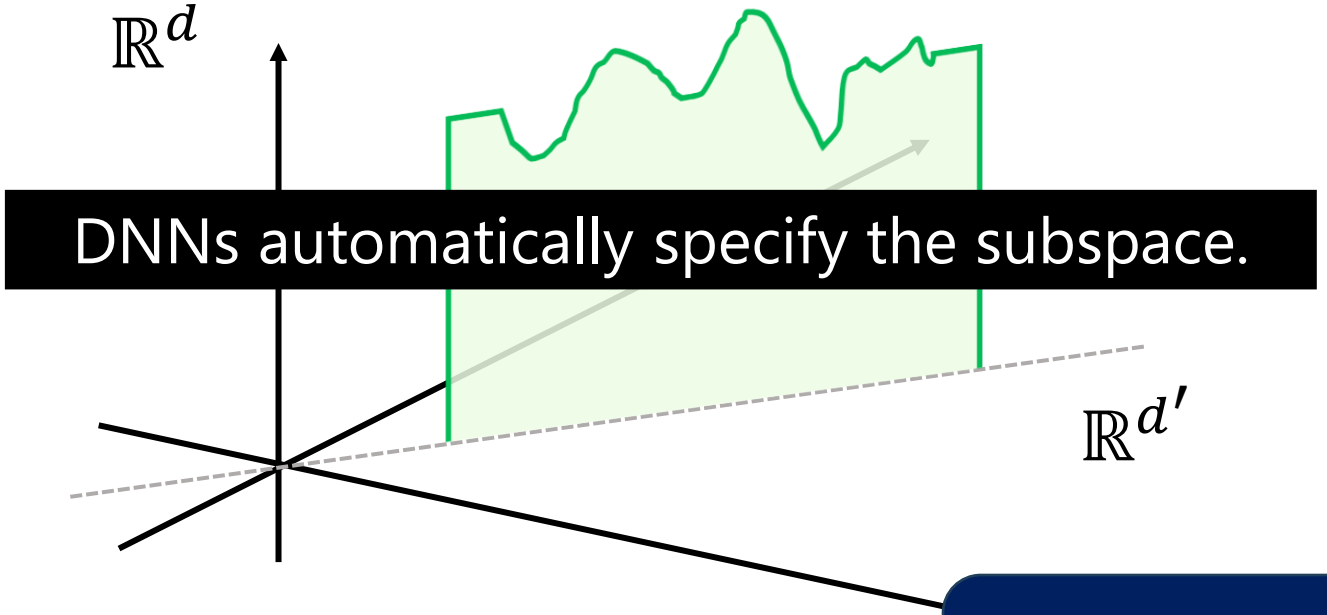
$$\mathbb{E}_{D_n}\left[\mathrm{TV}(\hat{Y}, X_0)\right] \lesssim n^{-\frac{\beta}{2\beta+d}} \log^9(n), \quad (\beta\text{: smoothness of density})$$

$$\mathbb{E}_{D_n}\left[\mathrm{W}_1(\hat{Y}, X_0)\right] \lesssim n^{-\frac{\beta+1-\delta}{2\beta+d'}} \quad (\text{for any } \delta > 0).$$

Both rates are (almost) minimax optimal [Yang & Barron, 1999; Niles-Weed & Berthet, 2022].

(Estimator for W_1 distance requires some modification)

Low dimensional manifold



Depends only on the intrinsic dimension

Theorem (Estimation error by W1-distance)

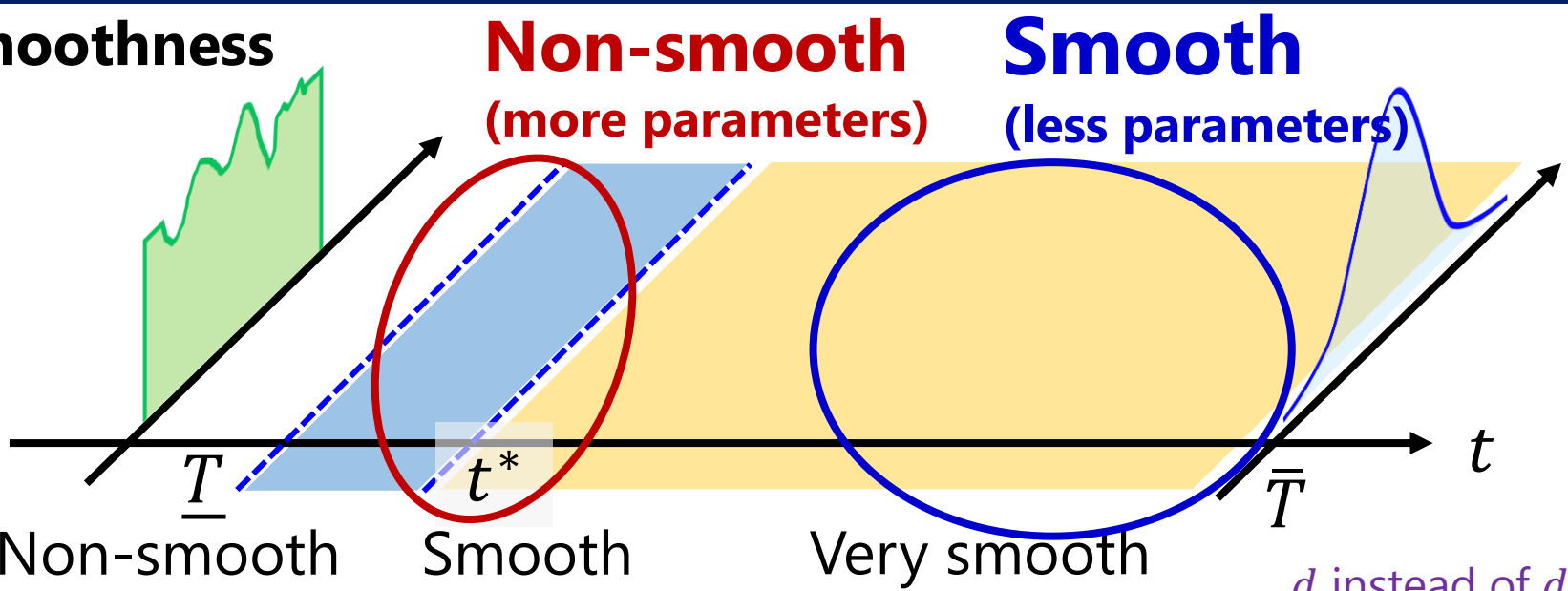
For any fixed $\delta > 0$, by slightly changing the estimat
the empirical risk minimizer $\hat{s}$ in DNN satisfies

$$\mathbb{E}_{D_n}\left[\mathrm{W}_1(\hat{Y}_{\overline{T}-\underline{T}}, X_0)\right] \lesssim n^{-\frac{\beta+1-\delta}{2\beta+d'}}. \qquad (d' > 2)$$

This is also known as minimax optimal (up to δ) [Niles-Weed & Berthet (2022)].

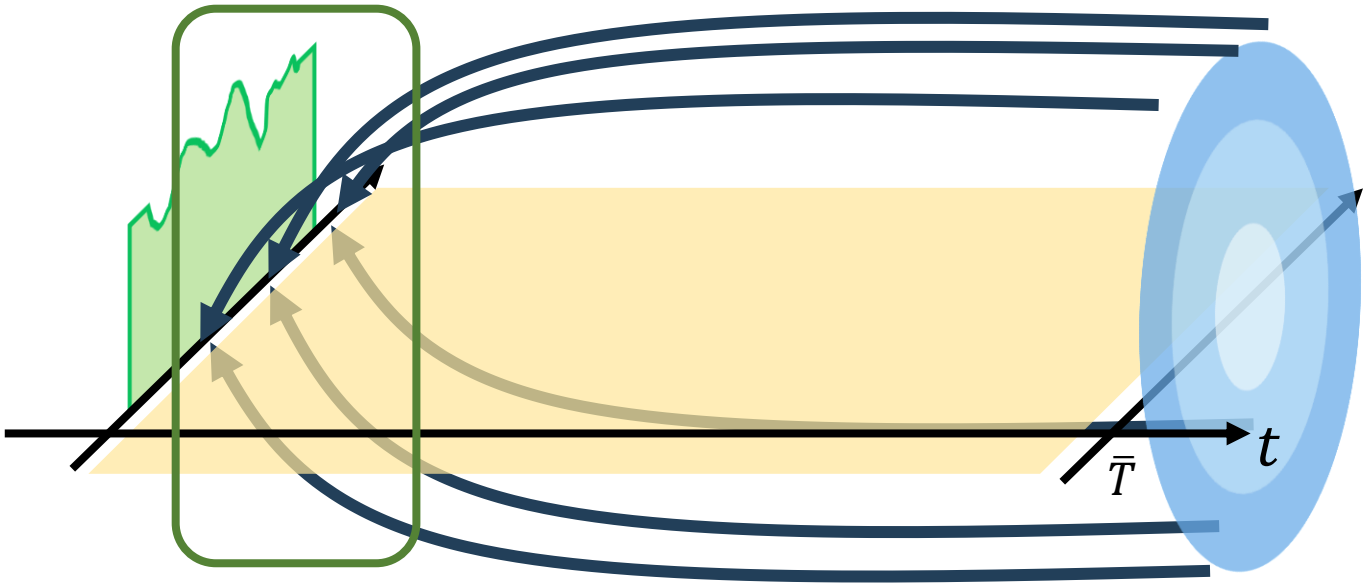
Two aspects of feature learning

- **Smoothness**



d instead of $d + 1$

- **Support identification**



d' instead of d

Two aspects of feature learning

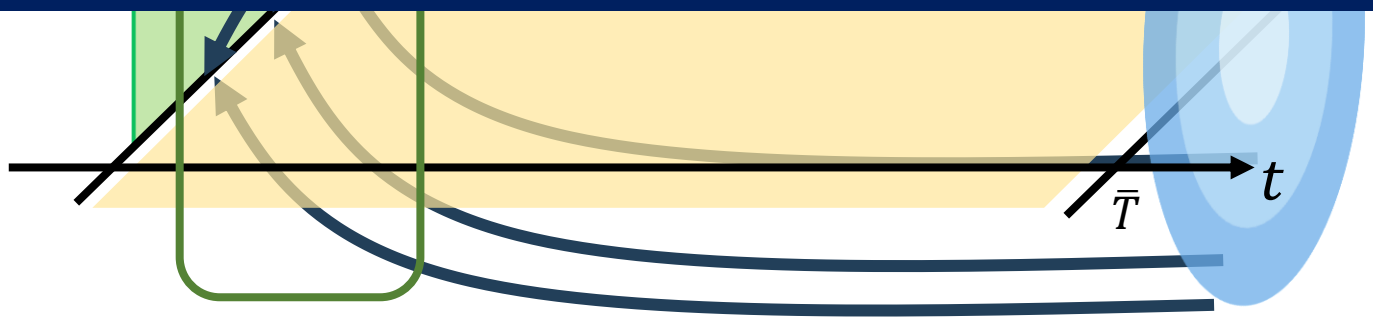
- **Smoothness**

Non-smooth (more parameters) **Smooth** (less parameters)

- The support is on a linear subspace.
  ⇒ Extend to curved manifold.
  [Fu, Suzuki, Lee, Nitanda, 2026]

- The data should be exactly on a subspace.
  ⇒ Extend to infinite dimension with anisotropic smoothness.
  [Maki, Suzuki, 2026]

$d + 1$

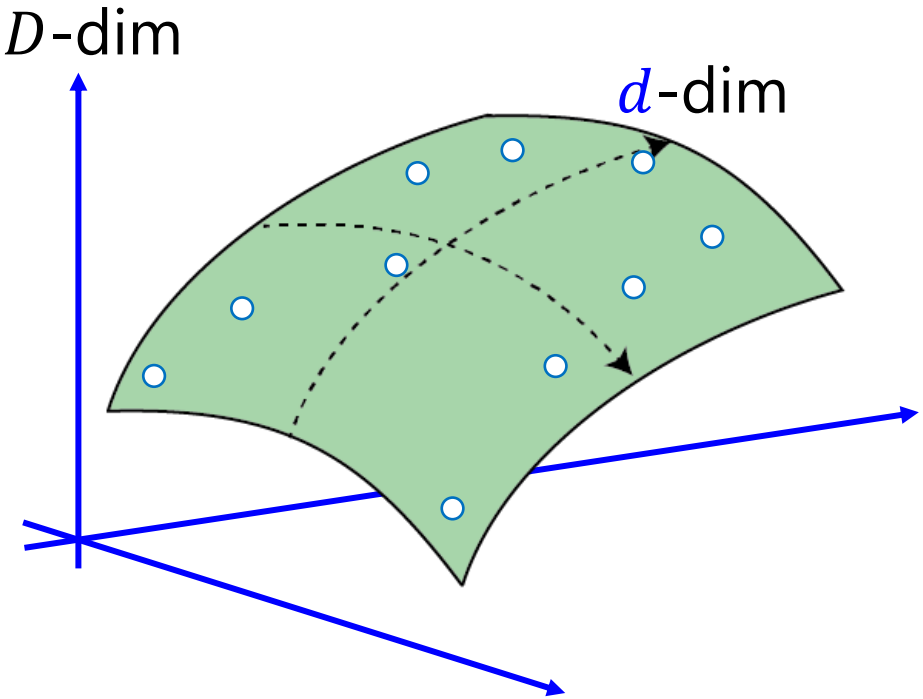


d' instead of d

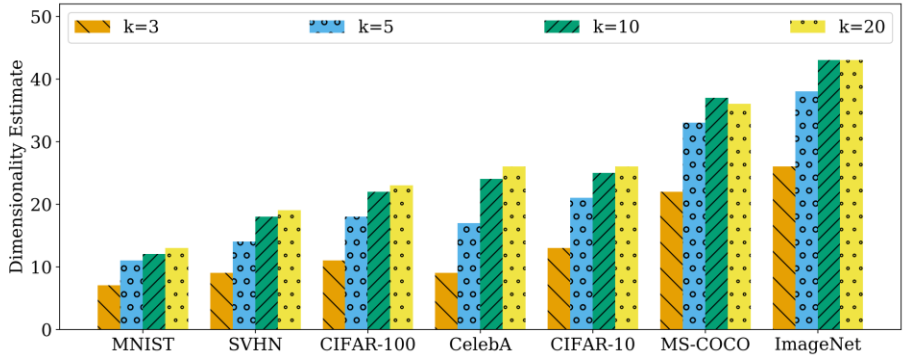
Fast rate with manifold assumption

[Guoji Fu, Taiji Suzuki, Wee Sun Lee, Atsushi Nitanda: Beyond the OU Process: Dimension-Adaptive Drifts for Minimax OptimalScore-Based Generative Modeling on Low-dimensional Manifolds. 2026]

Manifold assumption

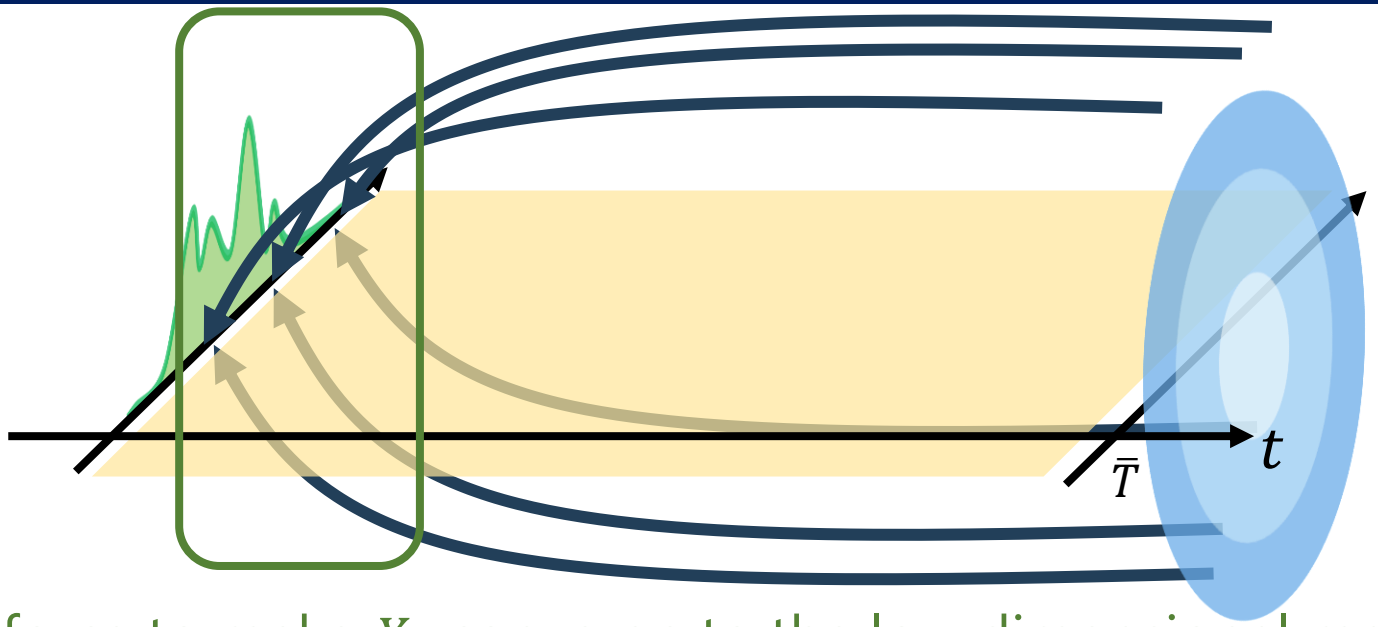


MNIST: 784 dim/13.4 intrinsic-dim [Facco et al. 2017]



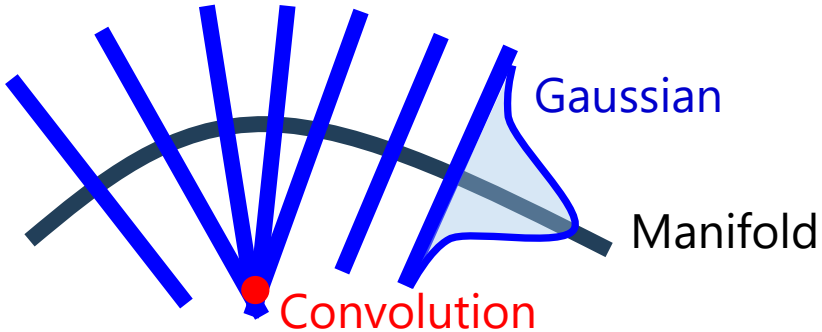
[Pope et al. ICLR2021]

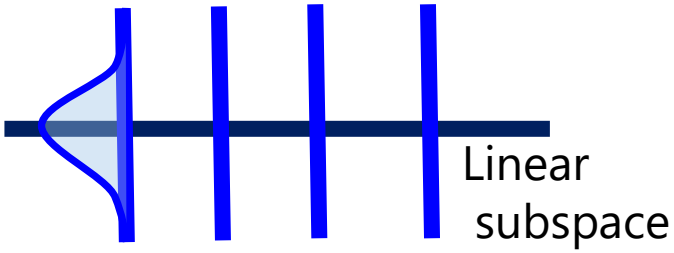
Difficulty



Strong force to make X_t converge to the low dimensional manifold.
⇒ Induces huge instability.

- If the manifold is a linear subspace, the orthogonal direction is just a Gaussian. → Easy to estimate.
- If the manifold is curved, there appears complicated convolution.
  → Difficult to estimate.





Relation to existing work

Table: Comparison of recent theoretical results on diffusion models for manifold data.

Paper	Manifold assumption	Density regularity	Metric	Convergence rate
Oko et al. (2023)	$\boldsymbol{X} = \boldsymbol{A}\boldsymbol{Z}$ $\mathrm{supp}(\boldsymbol{Z}) = [-1,1]^d$	Besov class density lower bounded density	W_1	$\tilde{\mathcal{O}}(n^{-\frac{\beta+1-\delta}{2\beta+d}})$
Chen et al. (2023)	$\boldsymbol{X} = \boldsymbol{A}\boldsymbol{Z}$ sub-Gaussian $P_{\boldsymbol{Z}}$	Lipschitz on-support score	TV	$\tilde{\mathcal{O}}(Dn^{-\frac{1-\delta}{d+5}})$ Sub-optimal
Tang and Yang (2024)	compact smooth manifold positive reach	Hölder class density lower bounded density	W_1	$\tilde{\mathcal{O}}(e^D n^{-\frac{\beta+1}{2\beta+d}})$
Azangulov et al. (2024)	compact smooth manifold positive reach	Hölder class density lower bounded density	W_1	$\tilde{\mathcal{O}}(\sqrt{D}n^{-\frac{\beta+1}{2\beta+d}})$
Yakovlev and Puchkin (2025)	$\boldsymbol{X} = g^*(\boldsymbol{Z}) + \boldsymbol{\xi}$ $\boldsymbol{Z} \sim \mathcal{U}([0,1]^d)$	Hölder class g^*	TV	$\tilde{\mathcal{O}}(D^{6+\binom{d+\beta}{d}}n^{-\frac{\beta}{6\beta+2d}})$ Sub-optimal
Our work	compact smooth manifold positive reach	Hölder class density	W_1	$\tilde{\mathcal{O}}(D^{d+7}n^{-\frac{\beta+1}{2\beta+d}})$

- Optimal rate w.r.t. W_1-metric with
  - **No lower bound assumption on the density.**
  - **Milder dependency on the ambient dimension.**

$$p_0 > C$$
$$\nabla \log(p_t) = \nabla p_t / p_t$$

Our contribution:

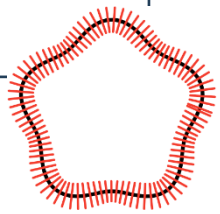
- Careful decomposition of the loss and novel time scheduling of the dynamics
  - Optimal rate without the lower bound assumption on the density.
  - Better dependency on the ambient dimension.

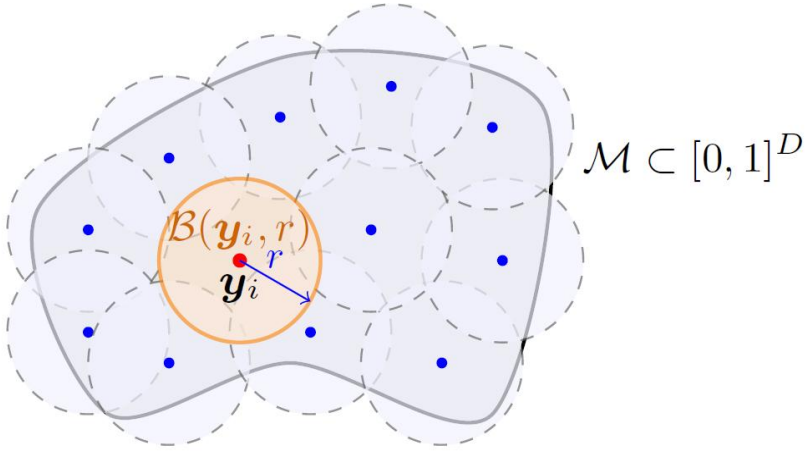
[**Concurrent work**: Zixuan Zhang, Kaixuan Huang, Tuo Zhao, Mengdi Wang, Minshuo Chen: Diffusion Model for Manifold Data: Score Decomposition, Curvature, and Statistical Complexity. arXiv:2603.20645]

Assumption

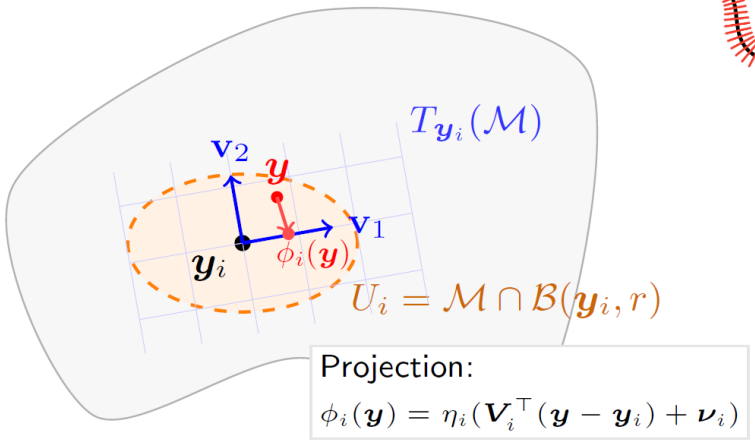
Assumption

- The data is supported on a smooth compact manifold $\mathcal{M}$ with a C^∞-atlas.
- $\exists$Density on the manifold and it is included in the β-Holder space.
- $\mathcal{M}$ is d-dimensional Riemannian manifold isometrically embedded in $\mathbb{R}^D$.
- $\mathcal{M}$ has a *reach* $\kappa > 0$ (critical radius).





Covering $\{\mathcal{B}(\boldsymbol{y}_i, r)\}_{i=1}^{C_\mathcal{M}}$ with radius $r < \kappa/2$



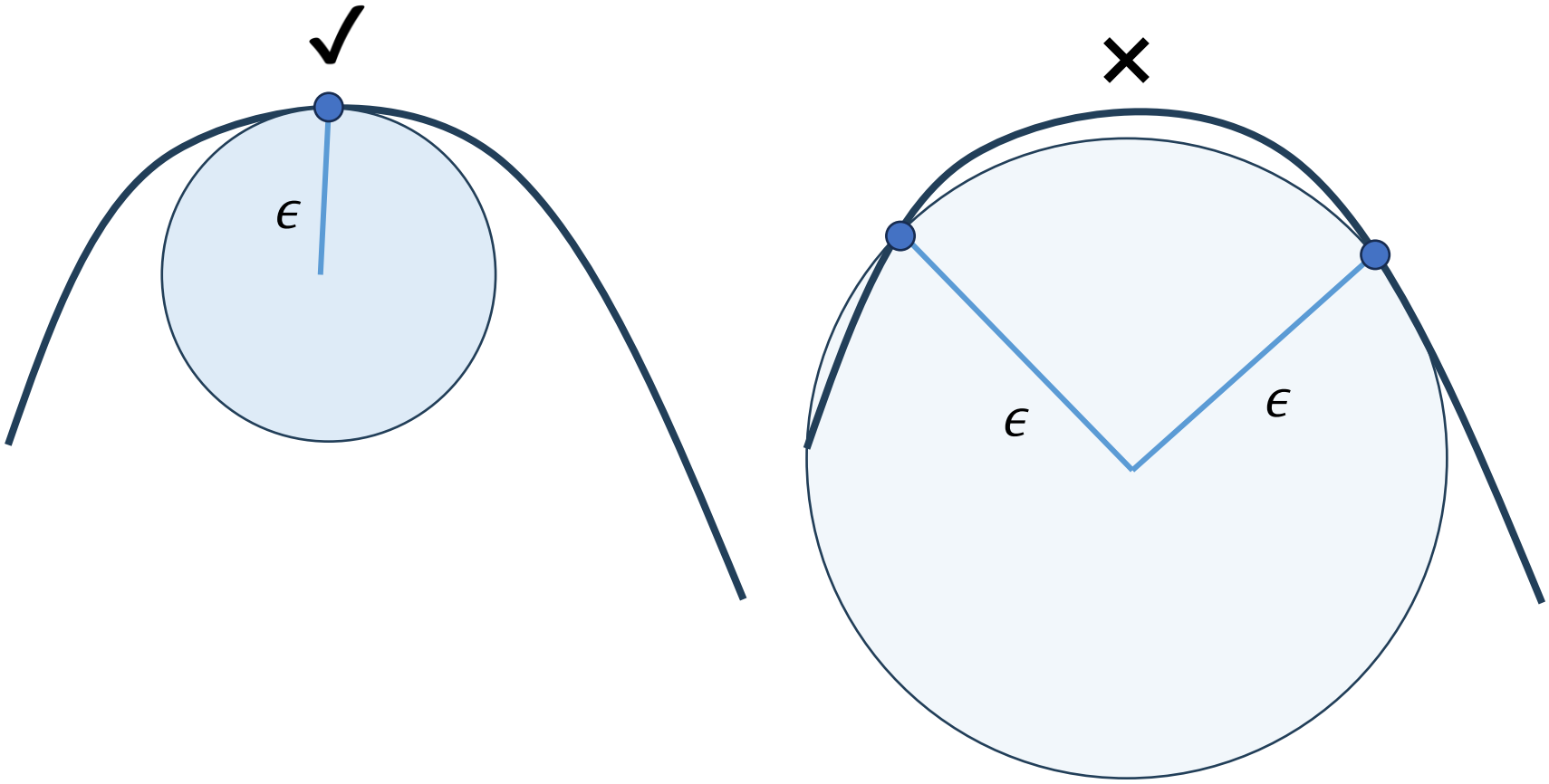
Good properties:

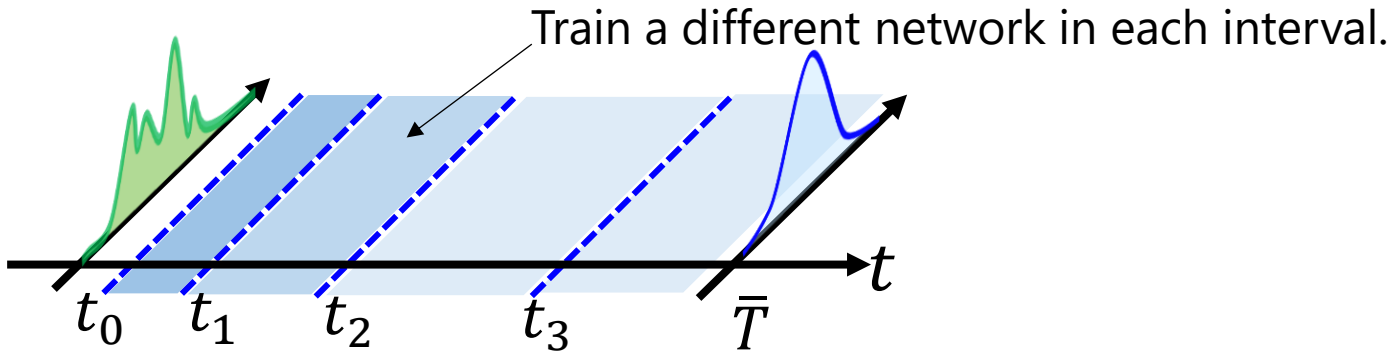
- Covering number bound: $N(r, \mathcal{M}) \lesssim r^{-d}$ $(\forall r \leq \kappa/2)$
- The local chart can be given as $\phi_i(y) = \eta_i(V_i^\top(y - y_i) + \nu_i)$ (projection onto the tangent space)
  - Can be exactly represented by a one-hidden layer ReLU network.

Definition of “reach”

$$\kappa := \sup\{\epsilon \mid \forall \epsilon \in M^{\epsilon},\ \exists! y \in M \text{ s.t. } \mathrm{dist}(x, M) = \|x - y\|\}$$

$$\text{where } M^{\epsilon} = \{x \in \mathbb{R}^D : \mathrm{dist}(x, M) < \epsilon\}$$





Theorem (Estimation error in W1-distance)

By carefully choosing the time step around $t = 0$ and training neural network at each small time interval, the DNN estimator $\hat{s}$ satisfies

$$\mathbb{E}_{(x_i)_{i=1}^n}\left[\mathrm{W}_1(\hat{P}_{t_0}, P_0)\right] \lesssim \begin{cases} n^{-1/2} D^9 \log^{10.5}(n) & (d \le 2), \\ n^{-\frac{\beta+1}{2\beta+d}} D^{d+7} \log^{\frac{5d}{4}+7}(n) & (d > 2). \end{cases}$$

⇒ Minimax optimal (up to polylog(n)) [Niles-Weed & Berthet (2022)].

If we train the network to minimize the averaged loss across the time, then we only have a naïve bound $n^{-\beta/(2\beta+d)}$.

Error decomposition

$$\mathbb{E}_{X_t\sim p_t}[\|\hat{s}(X_t,t)-\nabla\log p_t(X_t)\|^2]$$

$$\lesssim \underbrace{\int_{\mathrm{dist}(x_t,\mathcal{M})>\varrho_t}\|\hat{s}(x_t,t)-\nabla\log p_t(x_t)\|^2 p_t(x_t)\mathrm{d}x_t}_{(1)} + \underbrace{\int_{\mathrm{dist}(x_t,\mathcal{M})\le\varrho_t}\|\hat{s}(x_t,t)-\nabla\log p_t(x_t)\|^2 p_t(x_t)\mathrm{d}x_t}_{\text{Near the manifold}}$$

Away from the manifold

$\varrho_t \simeq \sigma_t\sqrt{\log(n)}$

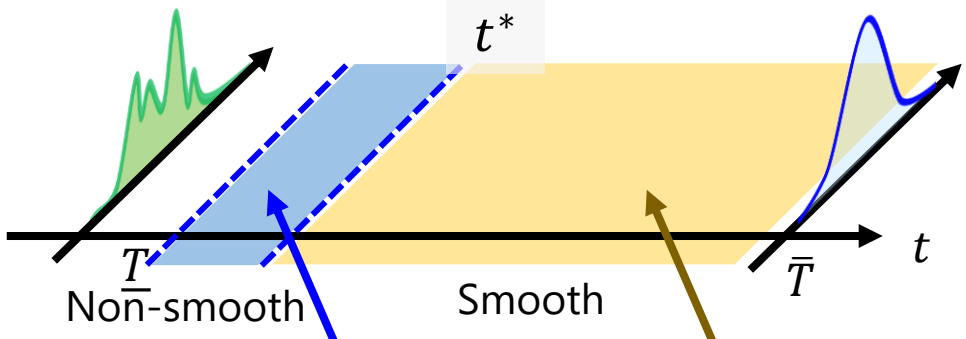
$O(n^{-O(1)})$

(2) $p_t(x_t) \le \rho_t$ (3) $p_t(x_t) > \rho_t$

$\rho_t = (2\pi)^{-\frac{D}{2}}(\sigma_t^2\log n)^{-\frac{D-d}{2}}\varepsilon,$

Decompose the error into two parts and balance them

(3)



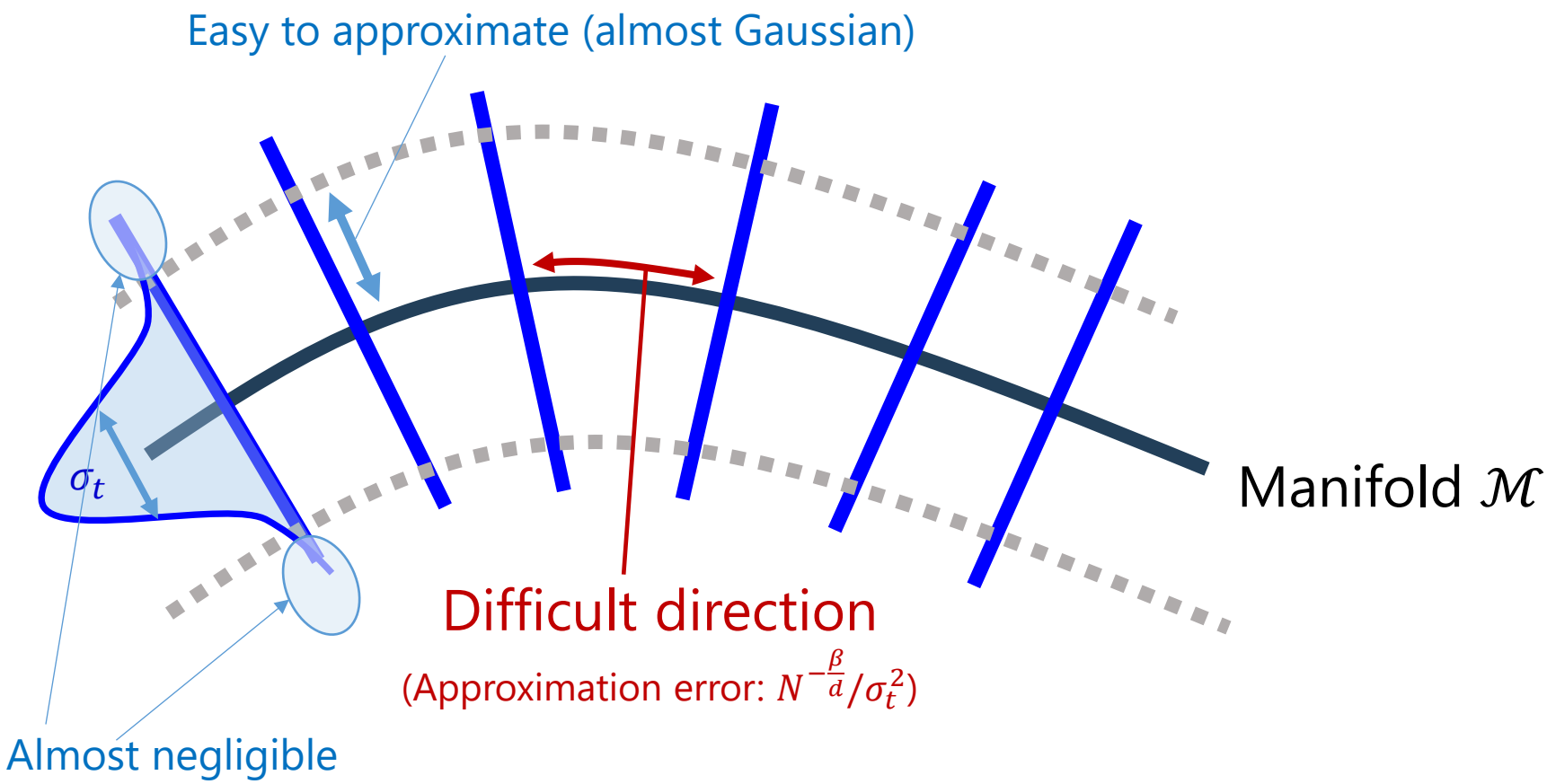
For small $t \le t_* = n^{-\frac{2}{\mathrm{d}+2\beta}}$,

$$\text{Error} \le \frac{N^{-\beta/d}}{\sigma_t^2} + \frac{N}{n}$$

with N parameter neural net

For larger $t \ge t_*$,

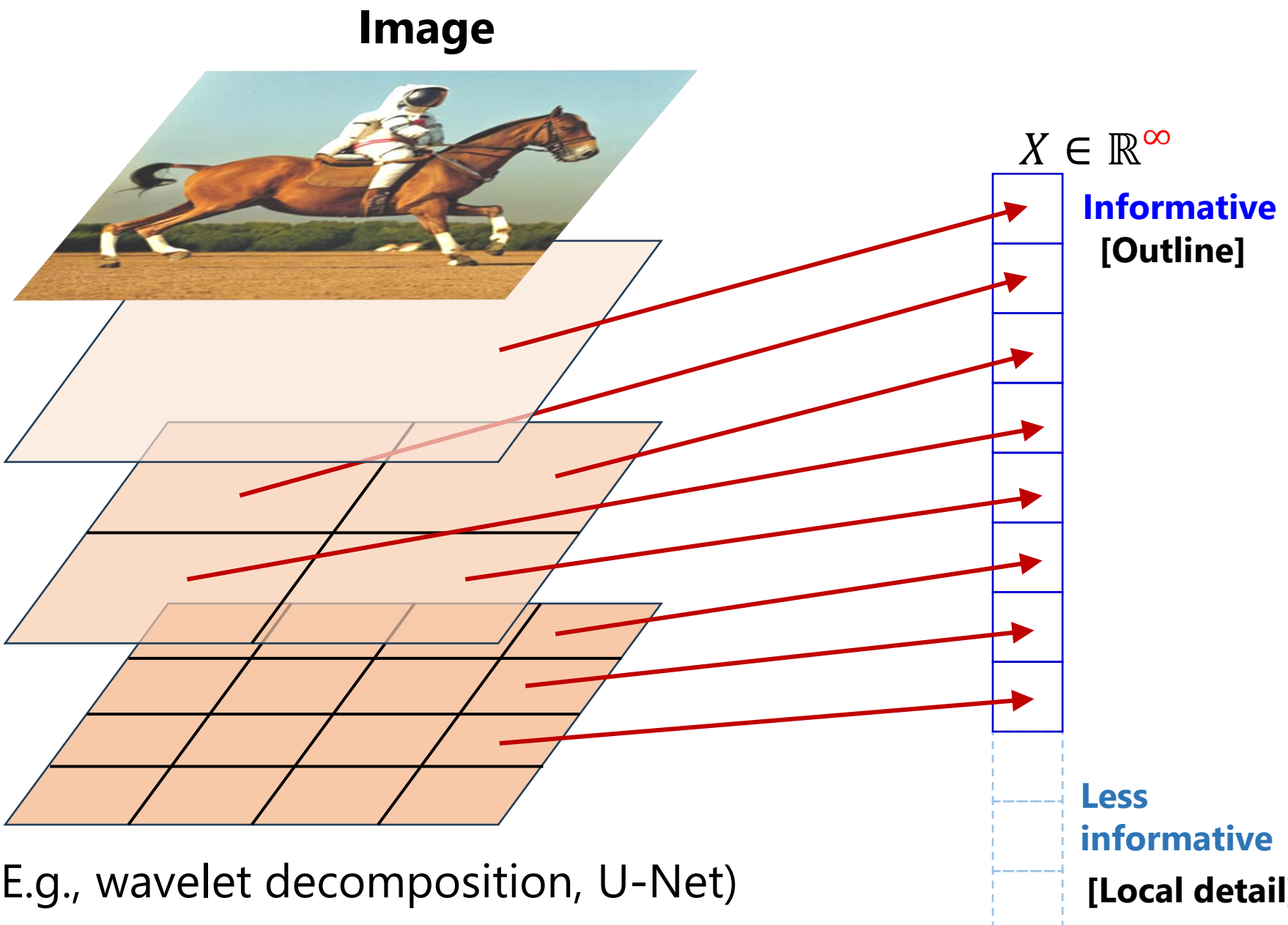
$$\text{Error} \le \frac{\sigma_t^{-d}}{n\sigma_t^2}$$


Easy to approximate (almost Gaussian)
σ_t
Manifold $\mathcal{M}$
Difficult direction
(Approximation error: $N^{-\frac{\beta}{d}}/\sigma_t^2$)
Almost negligible

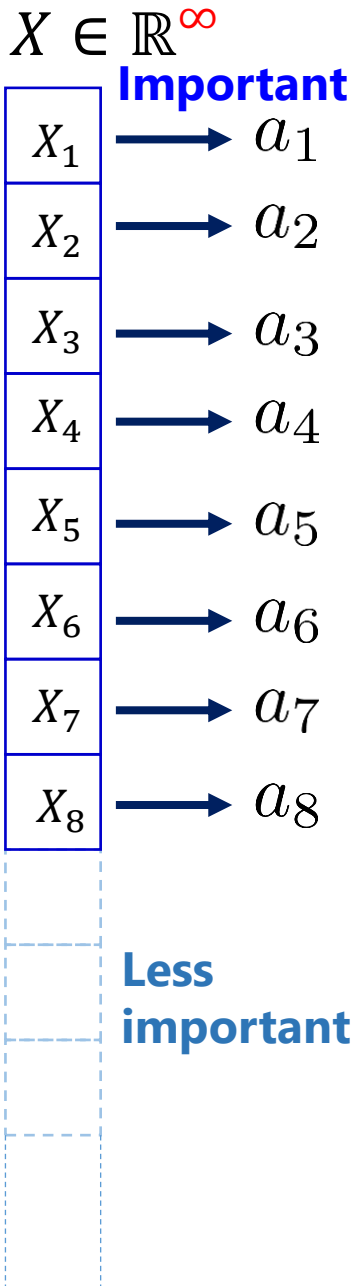
Infinite dimensional diffusion models

[Yuto Maki, Taiji Suzuki: Efficiency of Diffusion Models for Infinite-dimensional Data Generation: Dimension Independent Approximation and Estimation Errors. 2026]

Latent features



Anisotropic smoothness

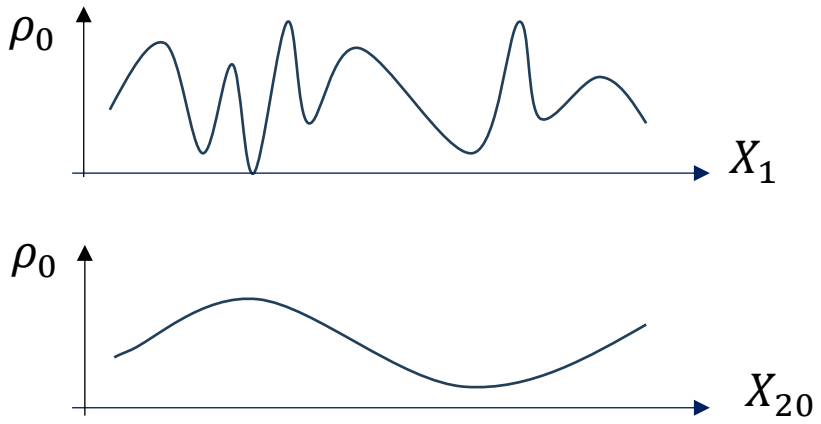


Each coordinate has its own **importance**.

(Important features have larger impact to the density)

Our approach:

Importance is characterized by **smoothness** w.r.t. the coordinate.



$$\left\| \frac{\partial^{a_i}}{\partial x_i^{a_i}} \left(\frac{\mathrm{d}P_0}{\mathrm{d}\mu_C} \right) \right\|_{L^p} < C$$

a_i times differentiable toward i-th coordinate.

Generalization error

$\hat{s} \in \underset{\check{s} \in \Phi(\hat{L}, \hat{W}, \hat{S}, \hat{B})}{\arg\min} \hat{R}(\check{s})$: Empirical risk minimizer in the DNN model.

Bounding the KL-divergence

$$\hat{R}(\check{s}) := \frac{1}{n} \sum_{i=1}^{n} \underset{t \sim \mathrm{Unif}[\underline{T}, \overline{T}]}{\mathbb{E}} [\| \check{s}(\tilde{x}_t, t) - 2C \nabla \log \tilde{p}_t(\tilde{x}_t | x_{0,i}) \|^2_{C^{-1}}]$$

- $\tilde{X}_t = X_{t,[1:d_{\mathrm{out}}]}$: the top d_{out} elements of X_t.
- $\tilde{p}_t$: distribution of $\tilde{X}_t$. Its conditional distribution $\tilde{p}_t(\cdot\,|x_0)$ is explicitly given.

Theorem (Generalization error)

$$v := \max\{1/p - 1/2, 0\} \quad \sigma_i(C) \leq i^{-2/r_0}$$

By setting the network size and d_{out} appropriately, we have that

$$\mathbb{E}[\mathcal{W}_1^2(P_0, \hat{Q}_{\underline{T}})] \lesssim n^{-\frac{(2-r_0)(a^\dagger - v)}{2(a^\dagger - v)+1}} \log^{5+\frac{1}{\eta}} n \log^3(\log n)$$

Anisotropic-smoothness

$$a^\dagger = \left(\sum_{i=1}^{\infty} a_i^{-1}\right)^{-1}$$

- Coordinate dependent smoothness yields polynomial order convergence even for infinite dimensional data.
- It matches the minimax optimal rate to estimate a γ-smooth function.

γ-smooth function class

ref

Let $\mathbb{Z}_0^\infty := \{I \in \mathbb{Z}^\infty \mid \mathrm{supp}(I) < \infty\}$,

$$\psi_{I_i}(x_i) = \begin{cases} \sqrt{2}\cos(2\pi I_i x_i) & (I_i > 0), \\ \sqrt{2}\sin(2\pi I_i x_i) & (I_i < 0), \\ 1 & (I_i = 0). \end{cases}$$

$$\psi_I(x) = \prod_{i=1}^{\infty} \psi_{I_i}(x_i) \quad \text{for } I \in \mathbb{Z}_0^\infty.$$

$$x = (x_i)_{i=1}^\infty \in [0,1]^\infty$$

Function class with infinite dimensional input that has different smoothness to each coordinate.

Def. (γ-smooth function class)

For $p > 1, q > 1, \gamma : \mathbb{N}_0^\infty \to \mathbb{R}$, define

$$\mathcal{F}_{p,q}^{\gamma}([0,1]^\infty) := \left\{ f = \sum_{I \in \mathbb{Z}_0^\infty} \alpha_I \psi_I \;\middle|\; \left(\sum_{s \in \mathbb{N}_0^\infty} 2^{q\gamma(s)} \|\delta_s(f)\|_p^q \right)^{1/q} < \infty \right\}$$

where $\delta_s(f)(x) = \sum \alpha_I \psi_I(x)$.

$s =$

$\gamma(s)$ represents a
n smoothness

Transform this function class to $\mathbb{R}^\infty$

- $\gamma(s) = \sum_{i=1}^{\infty} s_i a_i$: **mixed-smoothness**
- $\gamma(s) = \max_i \{s_i a_i\}$: **anisotropic-smoothness**

a_i represents smoothness toward the i-th coordinate.

$a_1 \le a_2 \le a_3 \le \cdots$

e)

Summary and conclusion

Part 1: Transformer (In-context learning)

Test time feature learning by softmax attention

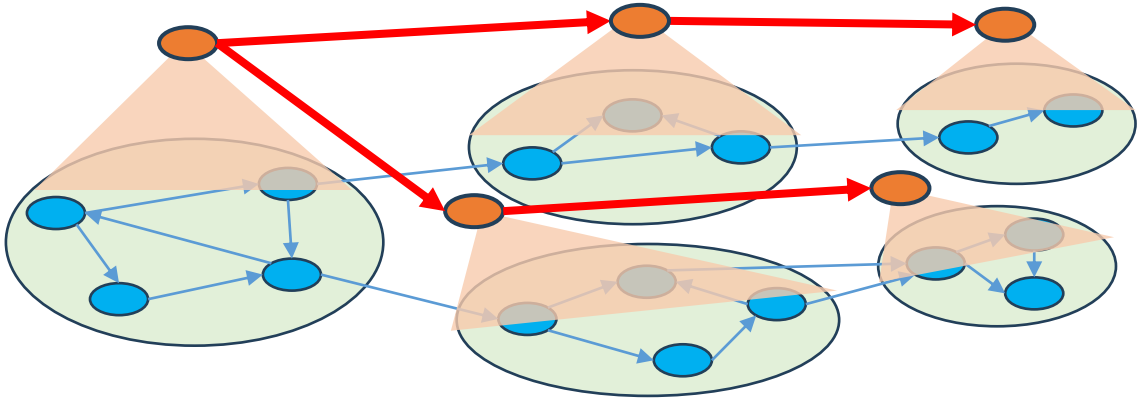
- Achieving the sample complexity with generative exponent
- Gaussian single index model

$r^{3\mathrm{ge}(\sigma^*)/2}$

Measure theoretic in-context learning

- Infinite-length context
- Measure to real-value problem

$$R(F^\star, \hat{F}_n) \lesssim \exp\{-\Omega((\log n)^{\alpha/(\alpha+1)})\}$$



Part 2: Diffusion models

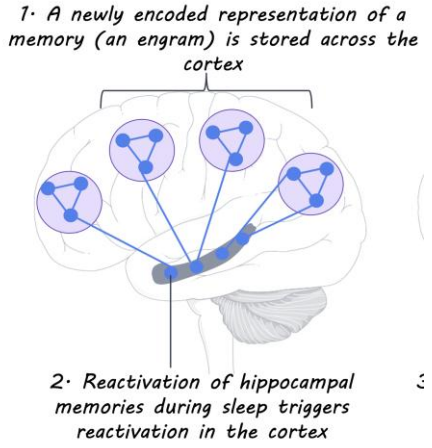
Minimax optimality

- Adaptation to smoothness
- Adaptation to low-dim subspace

General data support

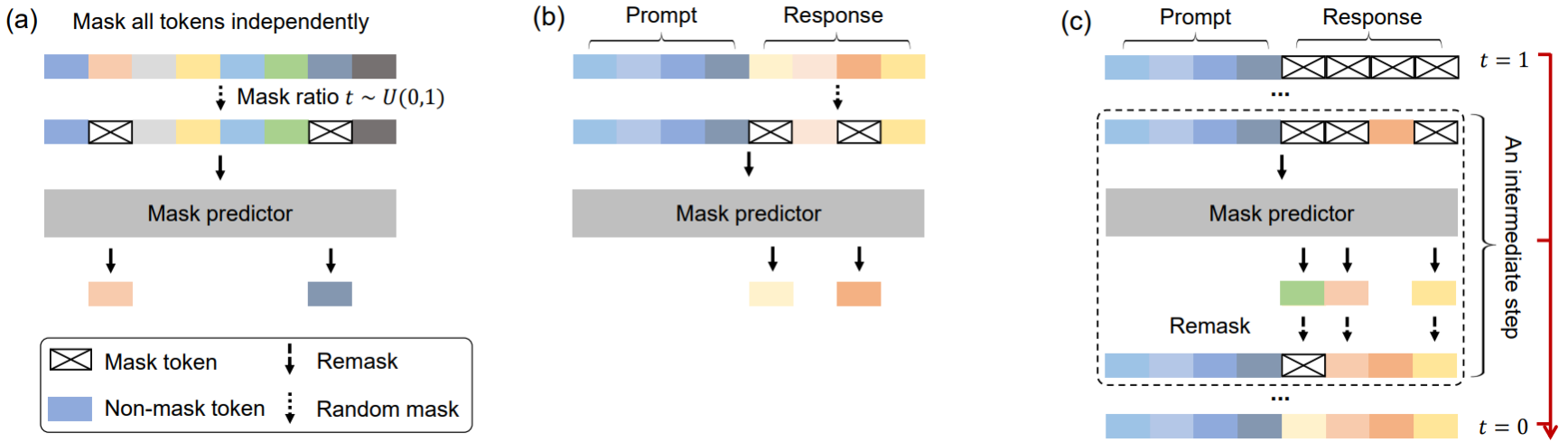
- Curved manifold
- Anisotropic smoothness

$$\mathbb{E}_{D_n}\left[\mathrm{W}_1(\hat{Y}, X_0)\right] \lesssim n^{-\frac{\beta+1-\delta}{2\beta+d'}}$$

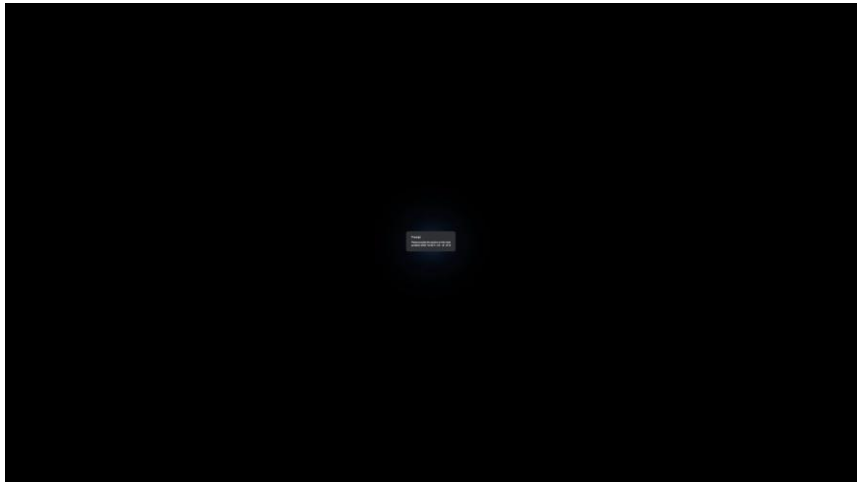


[BrainPost; 2021/1/26]

Discrete diffusion and language model

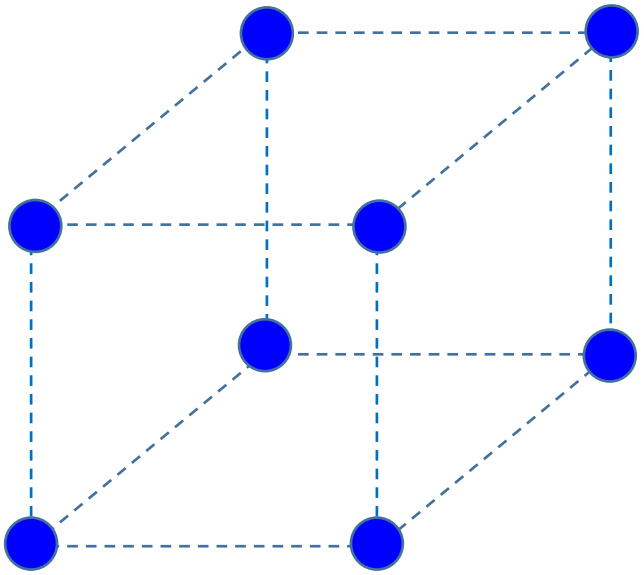


[Nie et al. Large Language Diffusion Models. arXiv:2502.09992]



[Google Deepmind: Gemini Diffusion, 2025]

Discrete data



$$\{1,2,\ldots,L\}^D$$

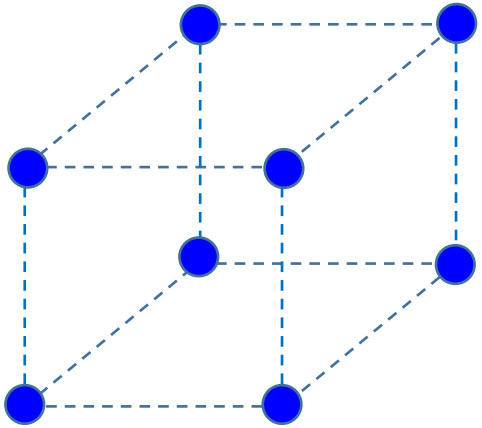
- The number of states exponentially grows w.r.t. the number of dimensions:

$$L^D$$

(curse of dimensionality)

Theory of discrete diffusion models

[Wakasugi, Suzuki: State Size Independent Statistical Error Bound for Discrete Diffusion Models. NeurIPS2025]



Example: learning a distribution on $\{0,1\}^D$

- Forward : $\frac{\mathrm{d}p_t}{\mathrm{d}t} = Qp_t, \qquad p_0 = p^*.$ (true distribution)
- Reverse : $\frac{\mathrm{d}q_t}{\mathrm{d}t} = \bar{Q}_{T-t}q_t \quad (t \in [0,T]), \quad q_0 = p_T.$

where $\bar{Q}_t(x,y) = \frac{p_t(y)}{p_t(x)}Q(x,y), \quad \bar{Q}_t(x,x) = -\sum_{y \neq x}\bar{Q}_t(x,y)$

Estimation as a multinomial distribution

$$D_{\mathrm{KL}}(p_0||\hat{q}_T) \lesssim \frac{2^D}{n}$$

Curse of dimensionality

Diffusion model by deep score model

Assumption

$p_0 = \sum_{j=1}^{M} c_j u_j(x)$ (eigen function decomposition)

u_j: j-th eigen function of Q

$\Rightarrow |c_j| \leq |c_1| j^{-s} \quad (s > 1)$

$$D_{\mathrm{KL}}(p_0||\hat{q}_T) \lesssim n^{-\frac{2(s-1)}{2(s-1)+1}}$$

Resolving curse of dimensionality

(only training important basis functions: feature learning)