LEARNING POLYNOMIAL TRANSFORMATIONS

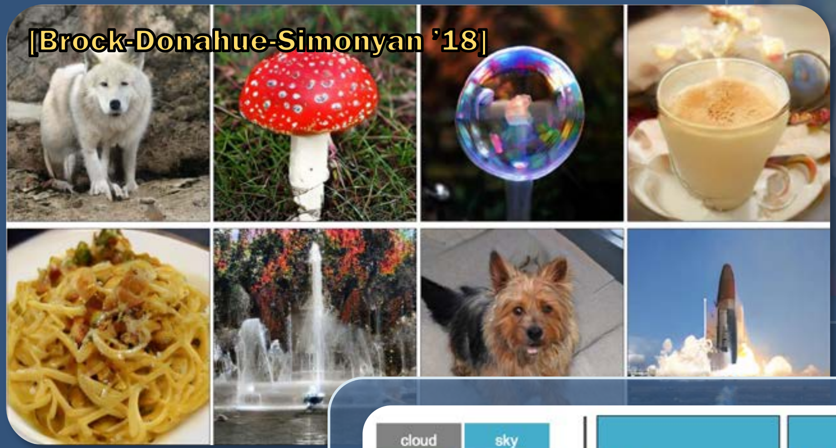
SITAN CHEN

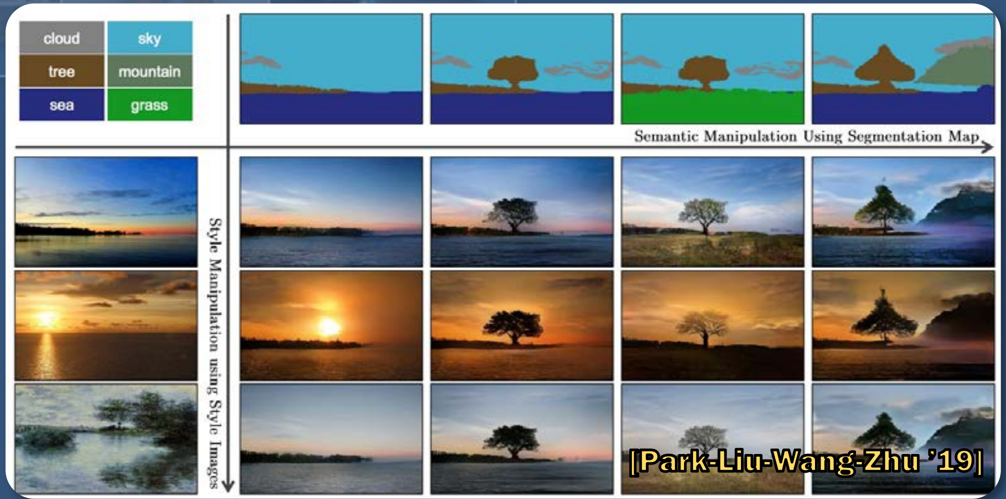
UC BERKELEY

JOINT W/ JERRY LI (MSR), YUANZHI LI (CMU), ANRU ZHANG (DUKE)

DEEP GENERATIVE MODELS

Powerful framework for modeling **real-world, high-dimensional distributions**

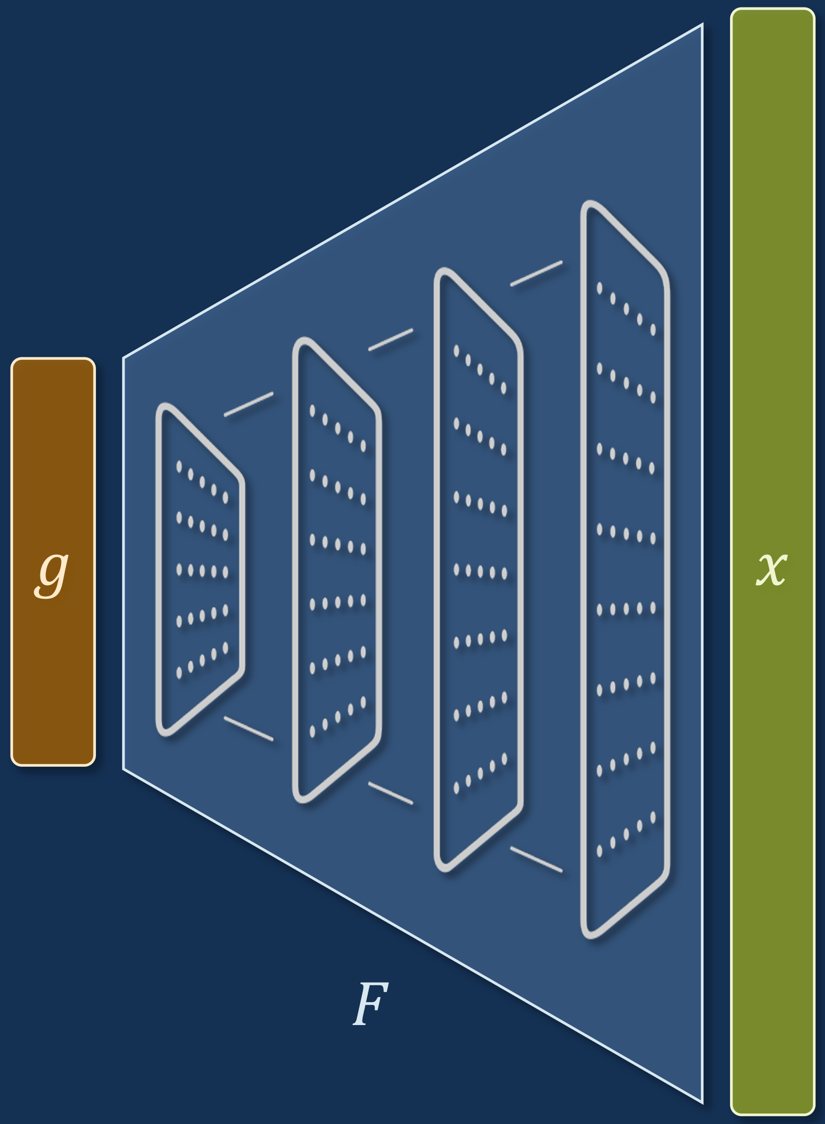




Idea: pushforwards of simple distributions, e.g. Gaussian, under **neural networks** are

1. **highly expressive**
2. (heuristically) **easy to optimize** over

DEEP GENERATIVE MODELS



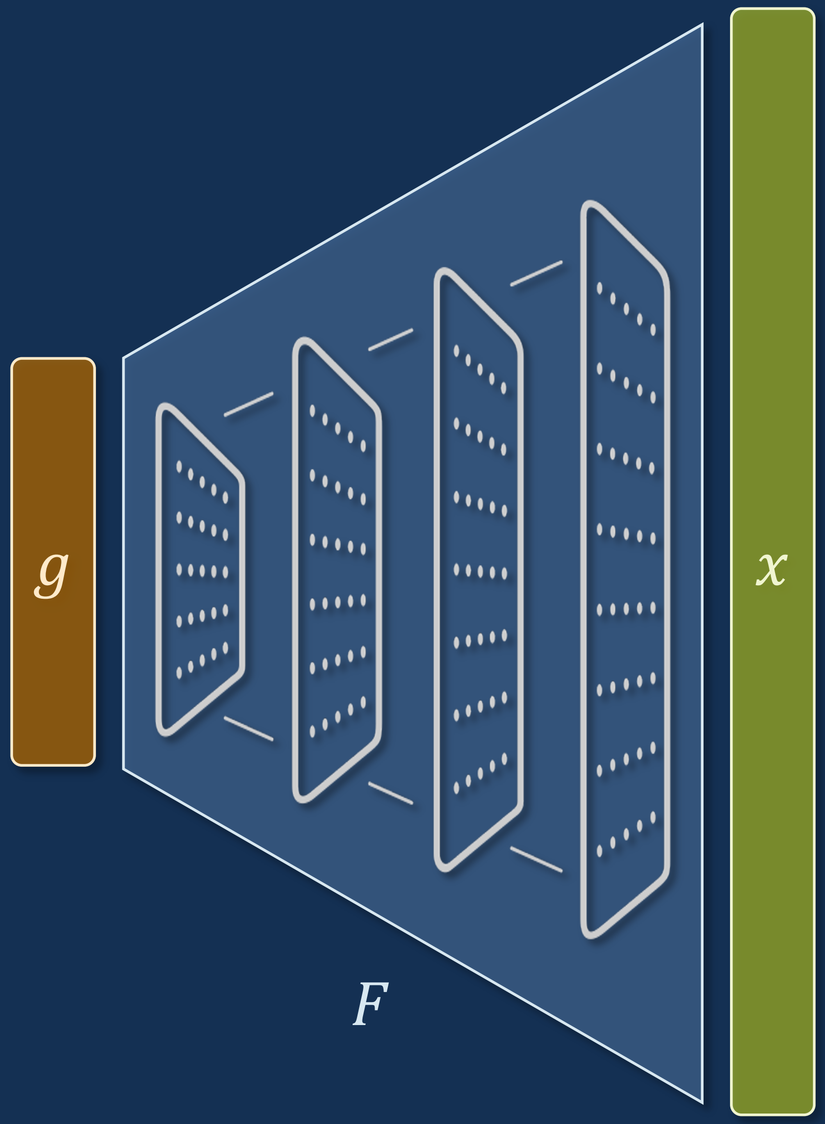
$F: \mathbb{R}^r \to \mathbb{R}^d$: parametric function (e.g. a neural network)

Goal: given samples from some real-world distribution D, produce F for which the **pushforward $F(N(\mathbf{0}, \mathbf{Id})) \approx D$**

To sample from $F\big(N(0, \mathrm{Id})\big)$:

1. Nature samples $g \sim N(0, \mathrm{Id})$
2. We observe $x = F(g)$

DEEP GENERATIVE MODELS



$F: \mathbb{R}^r \to \mathbb{R}^d$: parametric function (e.g. a neural network)

Goal: given samples from some real-world distribution D, produce F for which the **pushforward $F(N(\mathbf{0}, \mathbf{Id})) \approx D$**

E.g., GANs try to do this by solving

$$\min_F \max_A \left| \mathbb{E}_{x \sim D}[A(x)] - \mathbb{E}_{x \sim F(N(0,\mathrm{Id}))}[A(x)] \right|$$

"Discriminator"

MYSTERIES

Let's even say that D is exactly representable by a pushforward, i.e.
$\boldsymbol{D = F^*(N(0, \mathrm{Id}))}$ for some unknown neural net F^*

$$\min_F \max_A \left| \mathbb{E}_{x\sim D}[A(x)] - \mathbb{E}_{x\sim F(N(0,\mathrm{Id}))}[A(x)] \right|$$

Do heuristics for training GANs/VAEs successfully minimize these objectives?

In practice, expectations → sample averages, objective optimized by stochastic gradient descent-ascent (alternate between optimizing F and optimizing A)

Non-convex-concave objective, vanishing gradients, oscillations, mode collapse

[Arora-Ge-Liang-Ma-Zhang '17], [Arora-Zhang '17]: if A has bounded capacity, then even if training succeeds, $F(N(0, \mathrm{Id}))$ may have small support size

MYSTERIES

Let's even say that D is exactly representable by a pushforward, i.e.
$\boldsymbol{D} = \boldsymbol{F}^*(\boldsymbol{N}(\mathbf{0}, \mathbf{Id}))$ for some unknown neural net F^*

$$\min_F \max_A \left| \mathbb{E}_{x\sim D}[A(x)] - \mathbb{E}_{x\sim F(N(0,\mathrm{Id}))}[A(x)] \right|$$

Why do these objectives align with distribution learning?

[**C**-Li-Li-Meka '22]: For a large family of F^*'s, there exist F optimizing this objective, but for which **$\boldsymbol{D}$ and $\boldsymbol{F}(\boldsymbol{N}(\mathbf{0}, \mathbf{Id}))$ are far in Wasserstein distance**

(under a standard cryptographic assumption)

Idea: Consider $\boldsymbol{D} = \mathbf{Unif}(\{\pm \mathbf{1}\}^{\boldsymbol{d}})$. Pseudorandom generators transform $\boldsymbol{r} \ll \boldsymbol{d}$ **uniform bits** into distributions that are statistically far from D but "computationally" close to D.

MYSTERIES

Let's even say that D is exactly representable by a pushforward, i.e.
$\boldsymbol{D = F^*\left(N(0, \mathrm{Id})\right)}$ for some unknown neural net F^*

Is there *any* efficient algorithm that can provably learn such distributions?

In practice, no clear-cut way to evaluate how well a trained model has learned the distribution (unlike in supervised learning, e.g. test accuracy)

Various heuristics:

- **manually inspect** generated samples
  - compare them to nearest training images
  - take g, g' and evaluate F on the line between g, g'
- **heuristics for estimating log-likelihood** of held-out test data
- **Inception** score, Frechet **Inception** distance (based on pre-trained classifier)

MYSTERIES

Let's even say that D is exactly representable by a pushforward, i.e.
$\boldsymbol{D} = \boldsymbol{F}^*\left(\boldsymbol{N}(\mathbf{0}, \mathbf{Id})\right)$ for some unknown neural net F^*

Is there *any* efficient algorithm that can provably learn such distributions?

Long line of work in statistics and theoretical CS on **provable algorithms** for **learning high-dimensional distributions** from samples

Gaussian mixture models

[Dasgupta '99], [Vempala-Wang '04], [Moitra-Valiant '10], [Belkin-Sinha '10], [Hsu-Kakade '13], [Hardt-Price '15], [Ge-Huang-Kakade '15], [Regev-Vijayaraghavan '17], [Diakonikolas-Kane-Stewart '18], [Hopkins-Li '18], [Kothari-Steurer-Steinhardt '18], [Diakonikolas-Kane '20], [Diakonikolas-Hopkins-Kane-Karmalkar '20], [Bakshi-Kothari '20], [Liu-Moitra '21], [Li-Liu '21], [Bakshi-Diakonikolas-Jia-Kane-Kothari-Vempala '22], ….

Topic modeling

[Papadimitriou-Raghavan-Tamaki-Vempala '00], [Feldman-O'Donnell-Servedio '08], [Chaudhuri-Rao '08], [Arora-Ge-Moitra '12], [Blei '12] [Arora-Ge-Halpern-Mimno-Moitra-Sontag-Wu-Zhu '13], [Rabani-Schulman-Swamy '14], [Li-Rabani-Schulman-Swamy '15], [Ke-Wang '17], [Arora-Ge-Kannan-Moitra '16], [C-Moitra '19], [Gordon-Mazaheri-Schulman-Rabani '20], [Gordon-Schulman '22], …

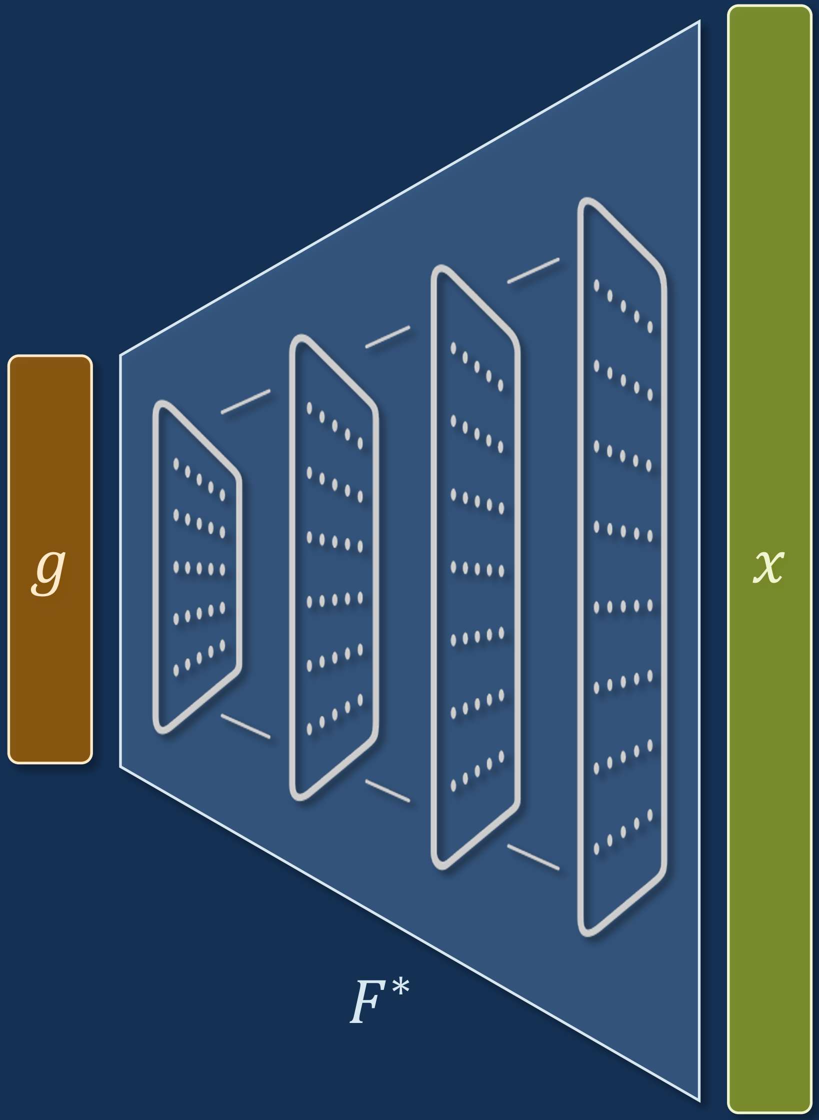
Graphical models

[Ravikumar-Wainwright-Lafferty '10], [Bresler-Gamarnik-Shah '14], [Bresler '15], [Vuffray-Misra-Lokhov-Chertkov '16], [Hamilton-Koehler-Moitra '17], [Meka-Klivans '17], [Kelner-Koehler-Meka-Moitra '19], [Bresler-Koehler-Moitra '19], [Wu-Sanghavi-Dimakis '19], [Goel-Kane-Klivans '19], [Bresler-Karzand '20], [Daskalakis-Pan '20], [Boix-Adsera-Bresler-Koehler '21], [Diakonikolas-Kane-Stewart-Sun '21],…

Independent Component Analysis

[Shalvi-Weinstein '90], [Comon '94], [Bell-Sejnowski '95], [Delfosse-Loubaton '95], [Frieze-Jerrum-Kannan '96], [Cardoso-Laheld '96], [Nguyen-Regev '06], [Vempala-Xiao '11], [Anandkumar-Foster-Hsu-Kakade-Liu '12], [Hsu-Kakade '12], [Arora-Ge-Moitra-Sachdeva '12], [Goyal-Vempala-Xiao '14], [Anderson-Goyal-Nandi-Rademacher '15], [Podosinnikova-Perry-Wein-Bach-d'Aspremont-Sontag '19], …

DEEP GENERATIVE MODELS

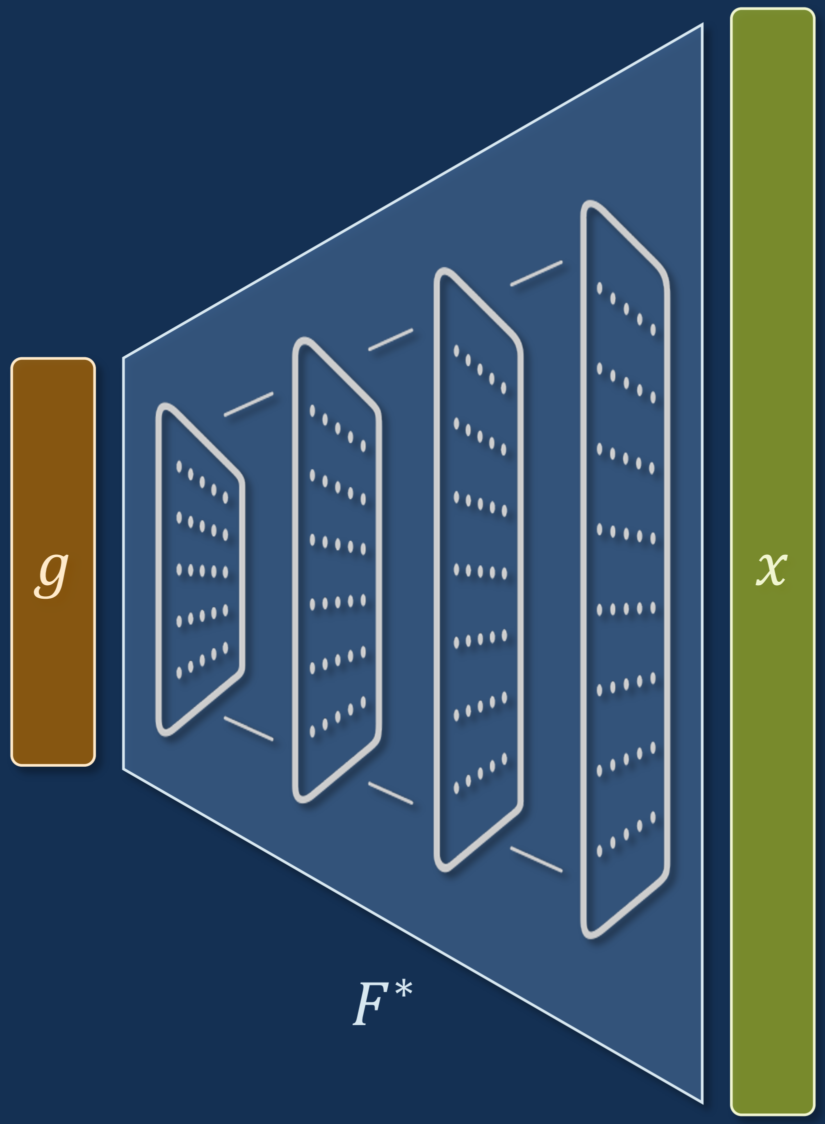


$F^*\colon \mathbb{R}^r \to \mathbb{R}^d$: unknown parametric function (e.g. a neural network)

Goal: given samples from $D = F^*\big(N(0, \mathrm{Id})\big)$, output F s.t. $F\big(N(0, \mathrm{Id})\big) \approx D$

e.g. Wasserstein

DEEP GENERATIVE MODELS



$F^*: \mathbb{R}^r \to \mathbb{R}^d$: unknown parametric function (e.g. a neural network)

Goal: given samples from $D = F^*\big(N(0, \mathrm{Id})\big)$, **recover the parameters of F^*** up to some error, modulo the natural symmetries

We will focus on case where each coordinate of F^* is **(homogeneous) low-deg. polynomial**

$$F^*(g) = \big(p_1(g), \ldots, p_d(g)\big)$$

"Depth-2 neural net with poly. activations"

PARAMETER RECOVERY

$F^*: \mathbb{R}^r \to \mathbb{R}^d$ given by degree-ω polynomials $p_1, \ldots, p_d: \mathbb{R}^r \to \mathbb{R}$

For now let's focus on $\omega = 2$

- Then can identify each p_a with symmetric matrix $Q_a \in \mathbb{R}^{r \times r}$

Gauge symmetry: for any $U \in O(r)$, $\{Q_a\}$ and $\{UQ_aU^\top\}$ give rise to the same pushforward distribution (b/c $N(0, \mathrm{Id})$ rotation-invariant)

- For $\omega = 2$, this is the only symmetry!

Goal: recover the parameters of $\boldsymbol{F}^*$ to within sufficient accuracy

PARAMETER RECOVERY

$F^*: \mathbb{R}^r \to \mathbb{R}^d$ given by degree-ω polynomials $p_1, \dots, p_d: \mathbb{R}^r \to \mathbb{R}$

For now let's focus on $\omega = 2$

Then can identify each p_a with symmetric matrix $Q_a \in \mathbb{R}^{r \times r}$

Gauge symmetry: for any $U \in O(r)$, $\{Q_a\}$ and $\{U Q_a U^\top\}$ give rise to the same pushforward distribution (b/c $N(0, \mathrm{Id})$ rotation-invariant)

For $\omega = 2$, this is the only symmetry!

Goal: output $\{\hat{Q}_a\}$ for which $\min_{U \in O(r)} \max_{1 \leq a \leq d} \left\| U Q_a U^\top - \hat{Q}_a \right\|_F \leq \varepsilon$

PARAMETER RECOVERY

$F^*: \mathbb{R}^r \to \mathbb{R}^d$ given by degree-ω polynomials $p_1, \ldots, p_d: \mathbb{R}^r \to \mathbb{R}$

~~For now let's focus on $\omega = 2$~~

Then can identify each p_a with symmetric matrix $Q_a \in \mathbb{R}^{r \times r}$

Gauge symmetry: for any $U \in O(r)$, $\{Q_a\}$ and $\{U Q_a U^\top\}$ give rise to the same pushforward distribution (b/c $N(0, \mathrm{Id})$ rotation-invariant)

For $\omega = 2$, this is the only symmetry!

Goal: output $\{\hat{Q}_a\}$ for which $\min_{U \in O(r)} \max_{1 \le a \le d} \left\| U Q_a U^\top - \hat{Q}_a \right\|_F \le \varepsilon$

PARAMETER RECOVERY

$F^*: \mathbb{R}^r \to \mathbb{R}^d$ given by degree-ω polynomials $p_1, \ldots, p_d: \mathbb{R}^r \to \mathbb{R}$

~~For now let's focus on $\omega = 2$~~

Then can identify each p_a with symmetric **tensor** $T_a \in (\mathbb{R}^r)^{\otimes \omega}$

Gauge symmetry: for any $U \in O(r)$, $\{Q_a\}$ and $\{U Q_a U^\top\}$ give rise to the same pushforward distribution (b/c $N(0, \mathrm{Id})$ rotation-invariant)

For $\omega = 2$, this is the only symmetry!

Goal: output $\{\hat{Q}_a\}$ for which $\min_{U \in O(r)} \max_{1 \le a \le d} \left\| U Q_a U^\top - \hat{Q}_a \right\|_F \le \varepsilon$

PARAMETER RECOVERY

$F^*: \mathbb{R}^r \to \mathbb{R}^d$ given by degree-ω polynomials $p_1, \ldots, p_d: \mathbb{R}^r \to \mathbb{R}$

~~For now let's focus on $\omega = 2$~~

Then can identify each p_a with symmetric **tensor** $T_a \in (\mathbb{R}^r)^{\otimes \omega}$

Gauge symmetry: for any $U \in O(r)$, $\{T_a\}$ and $\{F_U(T_a)\}$ give rise to the same pushforward distribution (b/c $N(0, \mathrm{Id})$ rotation-invariant)

For $\omega = 2$, this is the only symmetry!

$$F_U(T)_{i_1 \cdots i_\omega} = \sum_{j_1, \ldots, j_\omega} U_{i_1 j_1} \cdots U_{i_\omega j_\omega} T_{j_1 \cdots j_\omega}$$

Goal: output $\{\hat{Q}_a\}$ for which $\min_{U \in O(r)} \max_{1 \le a \le d} \left\| U Q_a U^\top - \hat{Q}_a \right\|_F \le \varepsilon$

PARAMETER RECOVERY

$F^*: \mathbb{R}^r \to \mathbb{R}^d$ given by degree-ω polynomials $p_1, \ldots, p_d: \mathbb{R}^r \to \mathbb{R}$

~~For now let's focus on $\omega = 2$~~

Then can identify each p_a with symmetric **tensor** $T_a \in (\mathbb{R}^r)^{\otimes \omega}$

Gauge symmetry: for any $U \in O(r)$, $\{T_a\}$ and $\{F_U(T_a)\}$ give rise to the same pushforward distribution (b/c $N(0, \mathrm{Id})$ rotation-invariant)

For $\omega = 2$, this is the only symmetry!

$$F_U(T)_{i_1 \cdots i_\omega} = \sum_{j_1, \ldots, j_\omega} U_{i_1 j_1} \cdots U_{i_\omega j_\omega} T_{j_1 \cdots j_\omega}$$

Goal: output $\{\hat{T}_a\}$ for which $\min_{U \in O(r)} \max_{1 \le a \le d} \left\| F_U(T_a) - \hat{T}_a \right\|_F \le \varepsilon$

PARAMETER RECOVERY

$F^*: \mathbb{R}^r \to \mathbb{R}^d$ given by degree-ω polynomials $p_1, \ldots, p_d : \mathbb{R}^r \to \mathbb{R}$

~~For now let's focus on $\omega = 2$~~

Then can identify each p_a with symmetric **tensor** $T_a \in (\mathbb{R}^r)^{\otimes \omega}$

Gauge symmetry: for any $U \in O(r)$, $\{T_a\}$ and $\{F_U(T_a)\}$ give rise to the same pushforward distribution (b/c $N(0, \mathrm{Id})$ rotation-invariant)

For $\omega > 2$, this is not the only symmetry!

[Grunbaum '75]: $p(x_1, x_2) = x_1^3 + x_1 x_2^2$, $q(x_1, x_2) = x_1^3 - 3x_1 x_2^2$

Goal: output $\{\hat{T}_a\}$ for which $\min_{U \in O(r)} \max_{1 \le a \le d} \left\| F_U(T_a) - \hat{T}_a \right\|_F \le \varepsilon$

WORST-CASE NETWORKS ARE HARD

Thm [C-Li-Li-Zhang]: Even for $d = 1, \omega = 2$, parameter recovery to $O(1)$ accuracy requires $\exp\big(\Omega(r)\big)$ samples in the worst case.

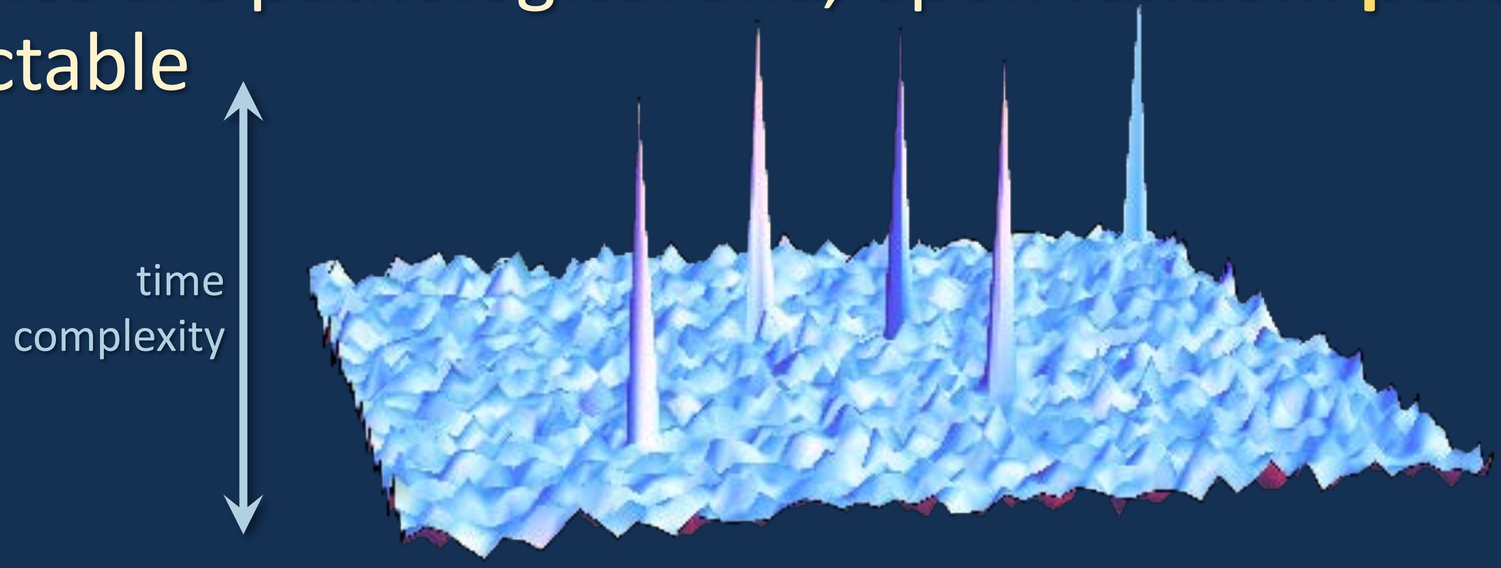
There exist pushforwards with very different parameters but which are exponentially close in statistical distance

Lower bound instance is delicate…real-world distributions will not look like this

Is learning / parameter recovery tractable for "non-worst-case" pushforwards?

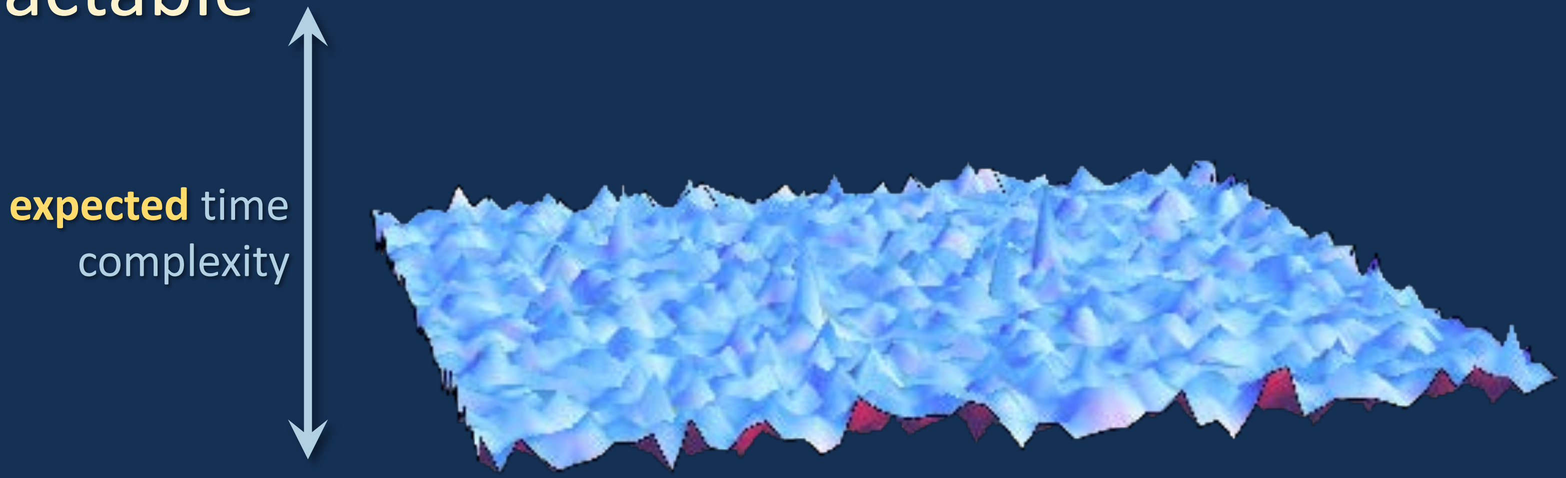
SMOOTHED ANALYSIS

Hard examples are pathological and, upon **random perturbation,** become tractable



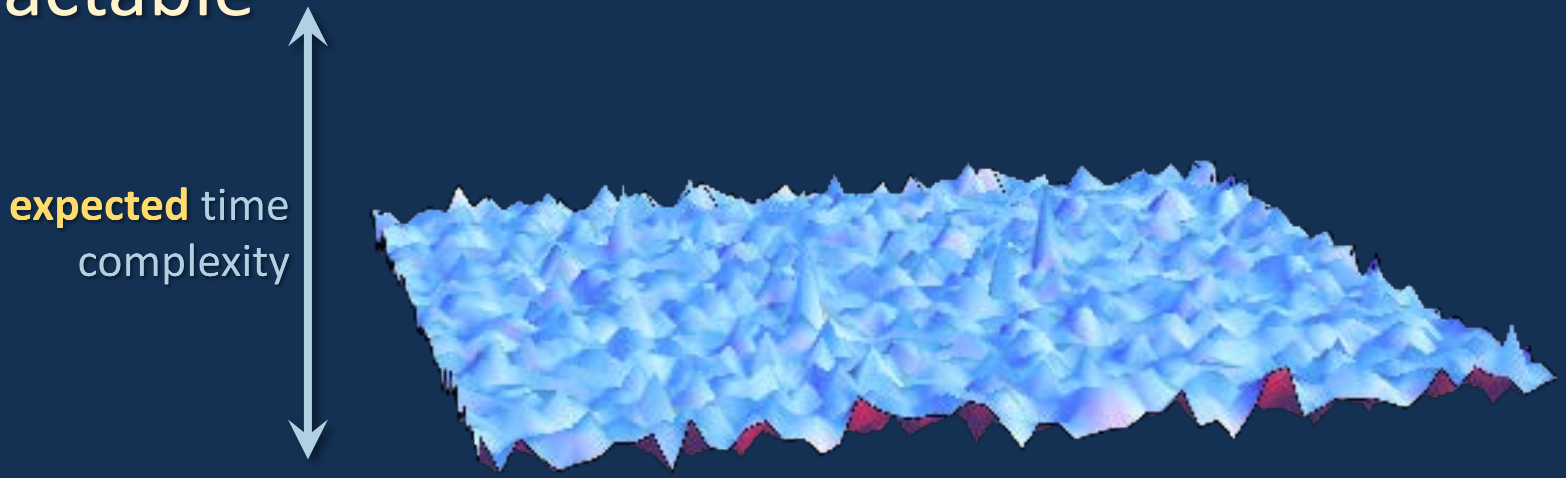
SMOOTHED ANALYSIS

Hard examples are pathological and, upon **random perturbation,** become tractable



SMOOTHED ANALYSIS

Hard examples are pathological and, upon **random perturbation,** become tractable



We consider "smoothed" $Q_1, \ldots, Q_d$ given by starting with any worst-case $\tilde{Q}_1, \ldots, \tilde{Q}_d$ and randomly perturbing each entry by $\sim 1/\mathrm{poly}(r)$

Note: more challenging than just considering "random" $\{Q_a\}$

PRELIMINARIES

Motivation

Setup

Results

QUADRATIC CASE

Moments and tensor ring

Proof of identifiability

Algorithm

HIGHER-DEGREE CASE

TAKEAWAYS

r: input dimension
d: output dimension
ω: degree of polynomial

OUR RESULTS

Thm [CLLZ]: For $\omega = 2$, there is an algorithm for recovering the parameters of any smoothed pushforward in time/samples $\mathbf{poly}(\boldsymbol{r}, \boldsymbol{d}, \mathbf{1}/\boldsymbol{\varepsilon})$ when $d \geq \Omega(r^2)$.

$d \geq \Omega(r^2)$ means distribution is supported on a low-dim. manifold

First end-to-end algorithmic result for learning a family of pushforwards computed by a neural network with > 1 layer

OUR RESULTS

r: input dimension
d: output dimension
ω: degree of polynomial

Thm [CLLZ]: For odd $\omega > 2$, there is an algorithm for recovering the parameters of any smoothed* push-forward in time/samples $\mathbf{poly}(\boldsymbol{r}, \boldsymbol{d}, \mathbf{1}/\boldsymbol{\varepsilon})$ when associated tensors $T_1, \ldots, T_d$ are rank-$\ell = O(1)$ and $d \geq \Omega\left(r^{c\omega\ell}\right)$.

*Because we focus on low-rank tensors, the smoothing here perturbs every rank-1 component individually, rather than the full tensors

When the polynomials are low-rank and non-worst-case, **gauge symmetry is the only symmetry, even for $\omega > 2$!**

PRELIMINARIES

Motivation

Setup

Results

QUADRATIC CASE

Moments and tensor ring

Proof of identifiability

Algorithm

HIGHER-DEGREE CASE

TAKEAWAYS

MOMENTS

If D is a degree-2 pushforward given by $Q_1, \ldots, Q_d \in \mathbb{R}^{r\times r}$, then

$$\frac{1}{2}\mathbb{E}_{z\sim D}[(z_a - \mathbb{E}[z_a])(z_b - \mathbb{E}[z_b])] = \mathbf{Tr}(\boldsymbol{Q_a Q_b})$$

Second-order moments tell us the angles between the Q's regarded as **$\boldsymbol{r^2}$-dimensional vectors**

This lets us recover them **up to a rotation in $\boldsymbol{O(r^2)}$**

To recover **up to a rotation in $\boldsymbol{O(r)}$**, must use **higher-order moments**

MOMENTS

If D is a degree-2 pushforward given by $Q_1, \dots, Q_d \in \mathbb{R}^{r\times r}$, then

$$\frac{1}{2}\mathbb{E}_{z\sim D}[(z_a - \mathbb{E}[z_a])(z_b - \mathbb{E}[z_b])] = \mathbf{Tr}(\boldsymbol{Q_a Q_b})$$

$$\frac{1}{8}\mathbb{E}_{z\sim D}[(z_a - \mathbb{E}[z_a])(z_b - \mathbb{E}[z_b])(z_c - \mathbb{E}[z_c])] = \mathbf{Tr}(\boldsymbol{Q_a Q_b Q_c})$$

Recovering $\{Q_a\}$ up to gauge symmetry from $\mathrm{Tr}(Q_a Q_b Q_c)$ is precisely the problem of **(symmetric) tensor ring decomposition**

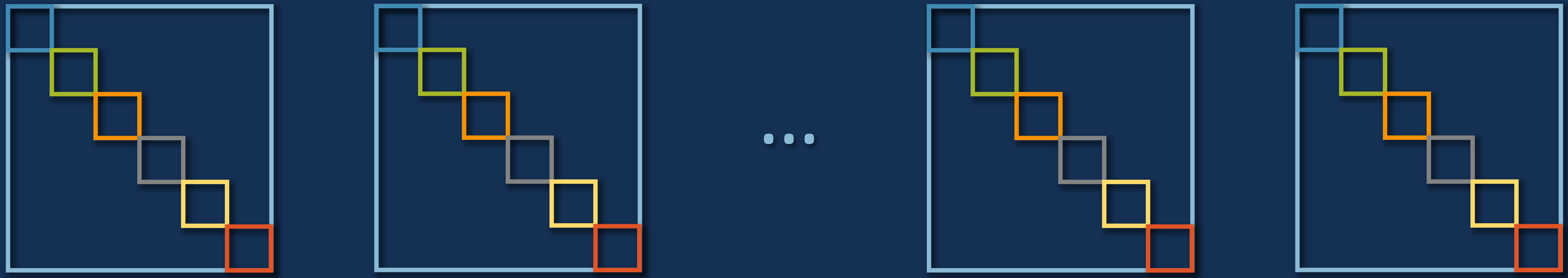
…equivalently, decomposing a translationally symmetric **matrix product state** of three particles with periodic boundary condition

MOMENTS

Recovering $\{Q_a\}$ up to gauge symmetry from $\mathrm{Tr}(Q_a Q_b Q_c)$ is precisely the problem of **(symmetric) tensor ring decomposition**

…equivalently, decomposing a translationally symmetric **matrix product state** of three particles with periodic boundary condition

Note: When Q_a's all diagonal, equivalent to **tensor decomposition**



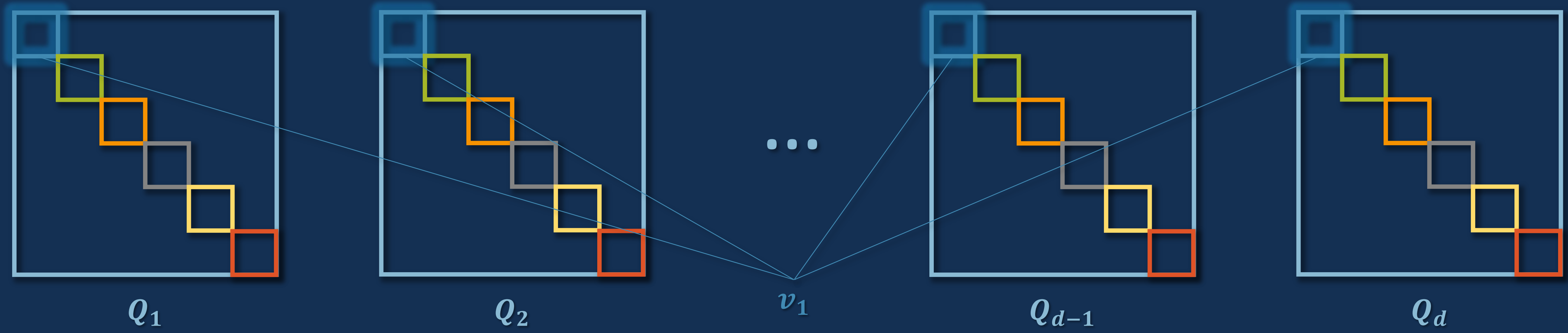
Q_1 Q_2 Q_{d-1} Q_d

MOMENTS

Recovering $\{Q_a\}$ up to gauge symmetry from $\mathrm{Tr}(Q_a Q_b Q_c)$ is precisely the problem of **(symmetric) tensor ring decomposition**

…equivalently, decomposing a translationally symmetric **matrix product state** of three particles with periodic boundary condition

Note: When Q_a's all diagonal, equivalent to **tensor decomposition**

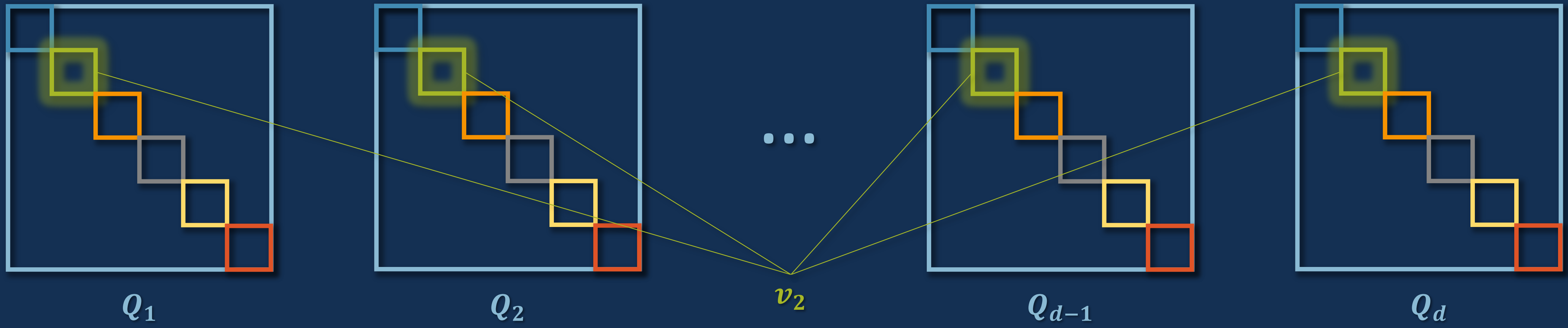


MOMENTS

Recovering $\{Q_a\}$ up to gauge symmetry from $\mathrm{Tr}(Q_a Q_b Q_c)$ is precisely the problem of **(symmetric) tensor ring decomposition**

...equivalently, decomposing a translationally symmetric **matrix product state** of three particles with periodic boundary condition

Note: When Q_a's all diagonal, equivalent to **tensor decomposition**

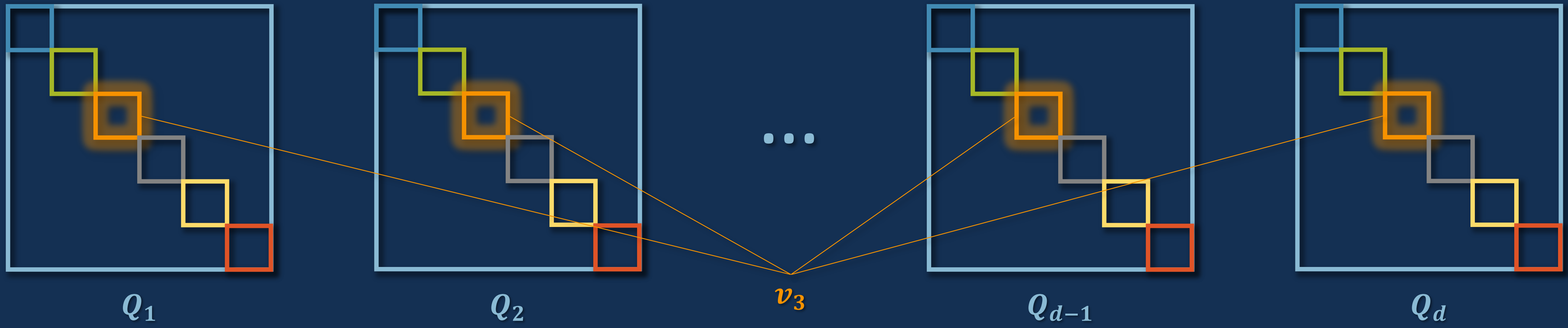


MOMENTS

Recovering $\{Q_a\}$ up to gauge symmetry from $\mathrm{Tr}(Q_a Q_b Q_c)$ is precisely the problem of **(symmetric) tensor ring decomposition**

...equivalently, decomposing a translationally symmetric **matrix product state** of three particles with periodic boundary condition

Note: When Q_a's all diagonal, equivalent to **tensor decomposition**

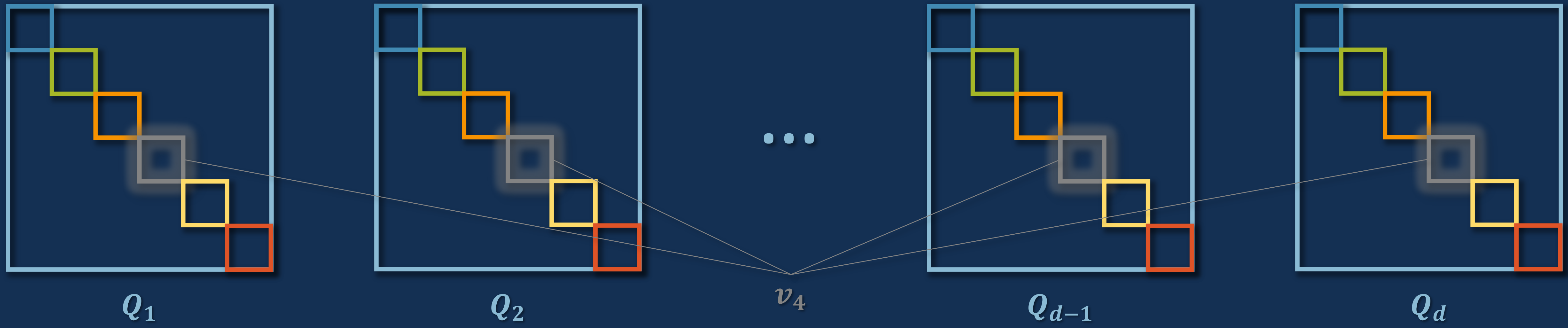


MOMENTS

Recovering $\{Q_a\}$ up to gauge symmetry from $\mathrm{Tr}(Q_a Q_b Q_c)$ is precisely the problem of **(symmetric) tensor ring decomposition**

…equivalently, decomposing a translationally symmetric **matrix product state** of three particles with periodic boundary condition

Note: When Q_a's all diagonal, equivalent to **tensor decomposition**

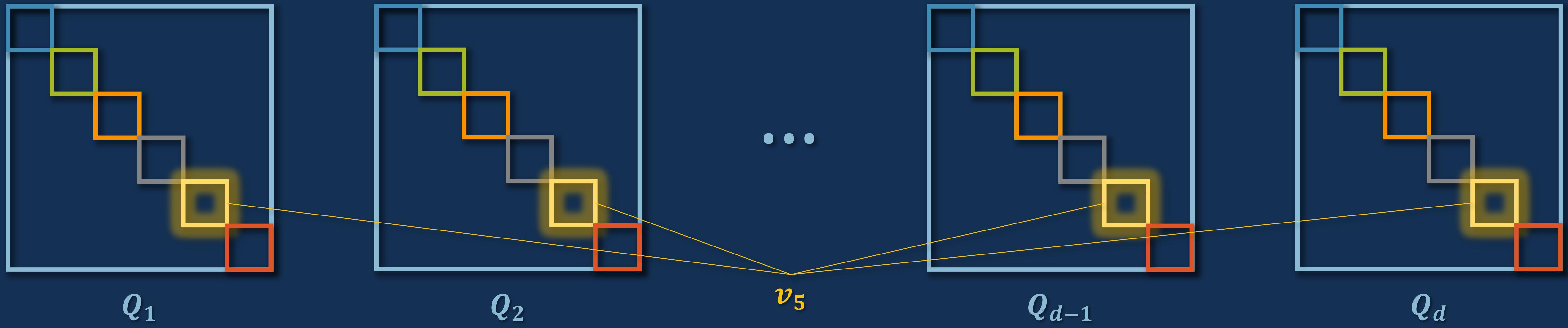


MOMENTS

Recovering $\{Q_a\}$ up to gauge symmetry from $\mathrm{Tr}(Q_a Q_b Q_c)$ is precisely the problem of **(symmetric) tensor ring decomposition**

…equivalently, decomposing a translationally symmetric **matrix product state** of three particles with periodic boundary condition

Note: When Q_a's all diagonal, equivalent to **tensor decomposition**

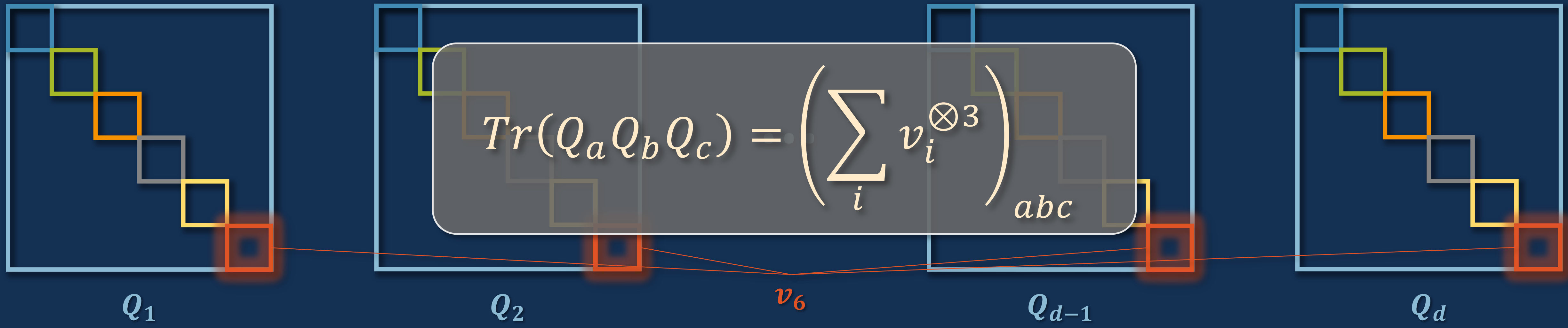


MOMENTS

Recovering $\{Q_a\}$ up to gauge symmetry from $\mathrm{Tr}(Q_a Q_b Q_c)$ is precisely the problem of **(symmetric) tensor ring decomposition**

…equivalently, decomposing a translationally symmetric **matrix product state** of three particles with periodic boundary condition

Note: When Q_a's all diagonal, equivalent to **tensor decomposition**



MOMENTS

Recovering $\{Q_a\}$ up to gauge symmetry from $\mathrm{Tr}(Q_a Q_b Q_c)$ is precisely the problem of **(symmetric) tensor ring decomposition**

…equivalently, decomposing a translationally symmetric **matrix product state** of three particles with periodic boundary condition

Note: When Q_a's all diagonal, equivalent to **tensor decomposition**

Widely studied, many provable algorithms known (Jennrich's algorithm, tensor power method, sum-of-squares, etc.)

When Q_a's are non-diagonal, no provable algorithms known!

IDENTIFIABILITY

A priori, not even clear that $\{Tr(Q_a Q_b)\}_{a,b}$ and $\{Tr(Q_a Q_b Q_c)\}_{a,b,c}$ uniquely determine $\{Q_a\}$ up to gauge symmetry!

Thm [CLLZ]: $\{\mathrm{Tr}(Q_a Q_b)\}_{a,b}$ and $\{\mathrm{Tr}(Q_a Q_b Q_c)\}_{a,b,c}$ uniquely and robustly determine $\{Q_a\}$ up to gauge symmetry when $d \geq \Omega(r^2)$ and $\{Q_a\}$ are smoothed.

i.e. if $Tr(Q_a Q_b) \approx Tr(\hat{Q}_a \hat{Q}_b)$ and $Tr(Q_a Q_b Q_c) \approx Tr(\hat{Q}_a \hat{Q}_b \hat{Q}_c)$ for all a, b, c, then there exists $U \in O(r)$ s.t. $U Q_a U^\top \approx \hat{Q}_a$ for all a

IDENTIFIABILITY

A priori, not even clear that $\{Tr(Q_a Q_b)\}_{a,b}$ and $\{Tr(Q_a Q_b Q_c)\}_{a,b,c}$ uniquely determine $\{Q_a\}$ up to gauge symmetry!

Thm [CLLZ]: $\{\mathrm{Tr}(Q_a Q_b)\}_{a,b}$ and $\{\mathrm{Tr}(Q_a Q_b Q_c)\}_{a,b,c}$ uniquely and robustly determine $\{Q_a\}$ up to gauge symmetry when $d \geq \Omega(r^2)$ and $\{Q_a\}$ are smoothed.

i.e., **the degree-≤ 3 moments of a quadratic pushforward "robustly identify" the parameters**

PRELIMINARIES

Motivation

Setup

Results

QUADRATIC CASE

Moments and tensor ring

Proof of identifiability

Algorithm

HIGHER-DEGREE CASE

TAKEAWAYS

PROOF OF IDENTIFIABILITY

Suppose $\{Q_a\}$ and $\{\hat{Q}_a\}$ satisfy

$$\mathrm{Tr}(Q_a Q_b) = \mathrm{Tr}(\hat{Q}_a \hat{Q}_b) \text{ for all } a, b$$

$$\mathrm{Tr}(Q_a Q_b Q_c) = \mathrm{Tr}(\hat{Q}_a \hat{Q}_b \hat{Q}_c) \text{ for all } a, b, c$$

Want to show there exists $U \in O(r)$ for which

$$U Q_a U^\top = \hat{Q}_a \text{ for all } a = 1, \dots, d$$

PROOF OF IDENTIFIABILITY

Recall: $\{\mathrm{Tr}(Q_a Q_b)\}_{a,b}$ specifies $\{Q_a\}$ **up to an $r^2 \times r^2$ rotation**

i.e. exists $W \in O(r^2)$ for which $W \underbrace{\mathrm{vec}(Q_a)}_{r^2 \times 1} = \mathrm{vec}(\hat{Q}_a)$

Note: if we had $U Q_a U^\top = \hat{Q}_a$ for all a, then $W = U \otimes U$

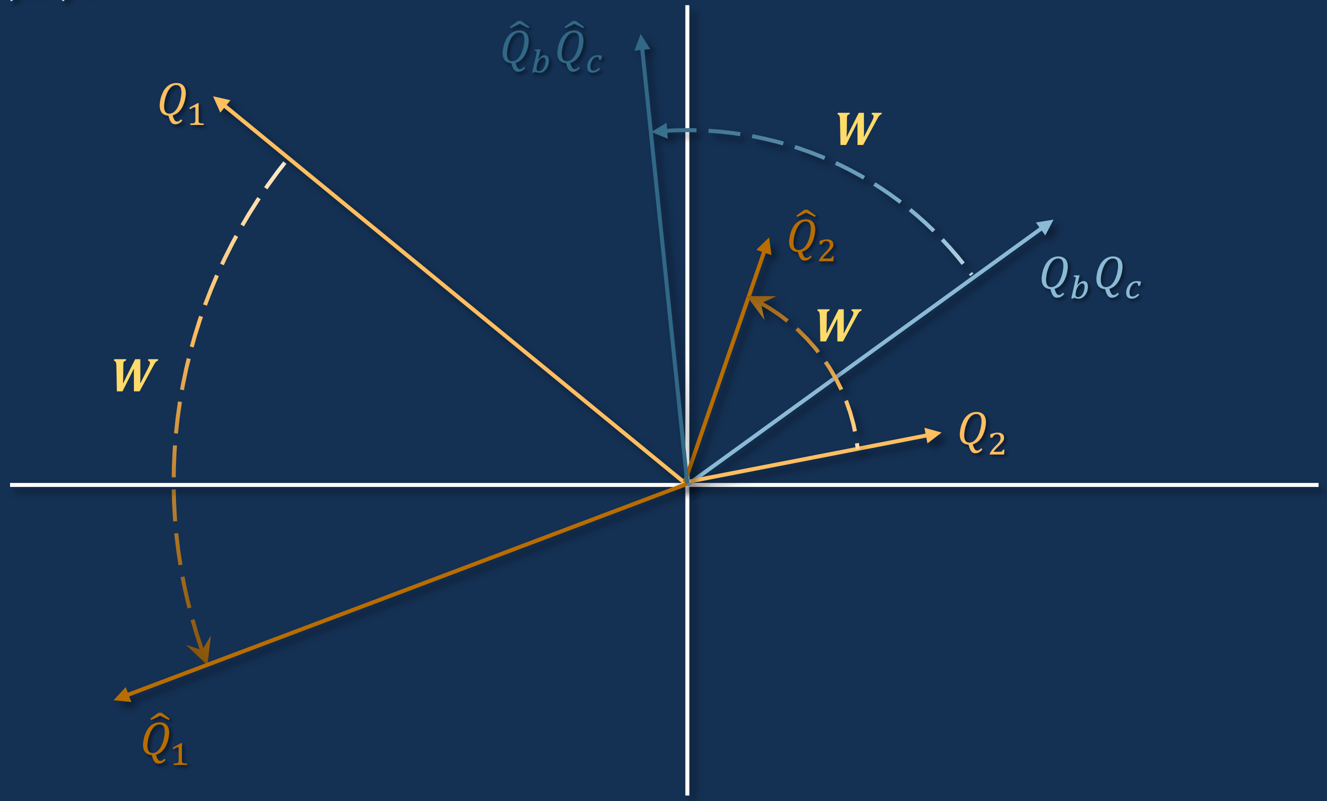
In this case, any column W^{ij} of W, regarded as an $r \times r$ matrix, would be rank-1, specifically $W^{ij} = U^i (U^j)^\top$

We will use $\{\mathrm{Tr}(Q_a Q_b Q_c)\}_{a,b,c}$ to prove W has this rank-1 structure

PROOF OF IDENTIFIABILITY

$\{\mathrm{Tr}(Q_a Q_b Q_c)\}_{a,b,c}$ tells us angles between every Q_a and every $Q_b Q_c$

$\left\{\mathrm{Tr}\left(\hat{Q}_a \hat{Q}_b \hat{Q}_c\right)\right\}_{a,b,c}$ tells us angles between every $\hat{Q}_a$ and every $\hat{Q}_b \hat{Q}_c$



PROOF OF IDENTIFIABILITY

$\{\mathrm{Tr}(Q_a Q_b Q_c)\}_{a,b,c}$ tells us angles between every Q_a and every $Q_b Q_c$

$\{\mathrm{Tr}(\hat{Q}_a \hat{Q}_b \hat{Q}_c)\}_{a,b,c}$ tells us angles between every $\hat{Q}_a$ and every $\hat{Q}_b \hat{Q}_c$

So W maps $Q_b Q_c$'s to $\hat{Q}_b \hat{Q}_c$'s (in addition to mapping Q_a's to $\hat{Q}_a$'s)

$$\boxed{\begin{aligned} W\,\mathrm{vec}(Q_a) &= \hat{Q}_a \qquad (1) \\ W\,\mathrm{vec}(Q_b Q_c) &= \hat{Q}_b \hat{Q}_c \qquad (2) \end{aligned}}$$

PROOF OF IDENTIFIABILITY

$$\boxed{\begin{aligned} W\ \mathrm{vec}(Q_a) &= \hat{Q}_a && (1) \\ W\ \mathrm{vec}(Q_b Q_c) &= \hat{Q}_b \hat{Q}_c && (2) \end{aligned}}$$

Suppose Q_a's consisted of $e_i e_j^\top$ for all $1 \leq i, j \leq r$

Then $(1) \rightarrow \hat{Q}_a$'s consist of $W\ \mathrm{vec}\left(e_i e_j^\top\right) = \underbrace{\text{columns } W^{ij}}_{r \times r}$

$$(2) \rightarrow \underbrace{W\ \mathrm{vec}\left(e_i e_j^\top \cdot e_k e_\ell^\top\right)}_{\Vert} = W^{ij} W^{k\ell}$$

$$W\ \mathrm{vec}\left(e_i e_\ell^\top\right) \cdot 1[j = k]$$

PROOF OF IDENTIFIABILITY

$$\boxed{\begin{aligned} W\,\mathrm{vec}(Q_a) &= \hat{Q}_a \qquad (1) \\ W\,\mathrm{vec}(Q_b Q_c) &= \hat{Q}_b \hat{Q}_c \qquad (2) \end{aligned}}$$

Suppose Q_a's consisted of $e_i e_j^\top$ for all $1 \le i, j \le r$

Then $(1) \to \hat{Q}_a$'s consist of $W\,\mathrm{vec}\left(e_i e_j^\top\right) = \underbrace{\text{columns } W^{ij}}_{r\times r}$

$$(2) \to \underbrace{W\,\mathrm{vec}\left(e_i e_j^\top \cdot e_k e_\ell^\top\right)}_{\parallel} = W^{ij} W^{k\ell}$$

$$W^{i\ell} \cdot 1[j = k]$$

PROOF OF IDENTIFIABILITY

$$\boxed{\begin{aligned} W\,\mathrm{vec}(Q_a) &= \hat{Q}_a \qquad (1) \\ W\,\mathrm{vec}(Q_bQ_c) &= \hat{Q}_b\hat{Q}_c \qquad (2) \end{aligned}}$$

Suppose Q_a's consisted of $e_ie_j^\top$ for all $1 \le i, j \le r$

$$\boxed{W^{ij}W^{k\ell} = W^{i\ell} \cdot 1[j = k] \qquad (*)}$$

For general $\{Q_a\}$, each identity of the form (2) yields some linear combination of identities of the form $(*)$

- With enough linearly independent $Q_a's$, these linear combinations span / imply the identities $(*)$

PROOF OF IDENTIFIABILITY

$$\boxed{W^{ij}W^{k\ell} = W^{i\ell} \cdot 1[j = k] \qquad (*)}$$

Claim: The columns of W (regarded as $r \times r$ matrices) are rank-1.

Proof:

$$\mathrm{Tr}\left(W^{ij}(W^{ij})^{\top}\right) = \mathrm{Tr}\left(W^{ii}W^{ij}(W^{ij})^{\top}\right) \qquad \text{by } (*)$$

$$\le \left\|W^{ii}\right\|_F \cdot \left\|W^{ij}(W^{ij})^{\top}\right\|_F$$

$$\ge$$

$$= \left\|W^{ij}(W^{ij})^{\top}\right\|_F \qquad (W \in O(r^2))$$

If trace of psd matrix $=$ Frobenius norm, it is rank-1.

PROOF OF IDENTIFIABILITY

$$\boxed{W^{ij}W^{k\ell} = W^{i\ell} \cdot 1[j = k] \qquad (*)}$$

Claim: $W = U \otimes U$ for $U \in O(r)$.

Proof: Say $W^{ij} = v_{ij}w_{ij}^{\top}$.

- $(W^{ii})^2 = W^{ii}$ implies $\boldsymbol{v}_{ii} = \boldsymbol{w}_{ii}$
- $W^{ii}W^{jj} = 1[i = j]$ implies $\{v_{11}, \ldots, v_{rr}\}$ **orthonormal**
- $W^{ii}W^{ij} = W^{ij}$ implies $\boldsymbol{v}_{ij} = \boldsymbol{v}_{ii}$
- $W^{ij}W^{jj} = W^{ij}$ implies $\boldsymbol{w}_{ij} = \boldsymbol{v}_{jj}$

So $W = U \otimes U$ where U's columns consist of $\{v_{11}, \ldots, v_{rr}\}$

PRELIMINARIES

Motivation

Setup

Results

QUADRATIC CASE

Moments and tensor ring

Proof of identifiability

Algorithm

HIGHER-DEGREE CASE

TAKEAWAYS

BREAKING SYMMETRY

Thm [CLLZ]: $\{\mathrm{Tr}(Q_a Q_b)\}_{a,b}$ and $\{\mathrm{Tr}(Q_a Q_b Q_c)\}_{a,b,c}$ uniquely and robustly determine $\{Q_a\}$ up to gauge symmetry when $d \geq \Omega(r^2)$ and $\{Q_a\}$ are smoothed.

Corollary: when $d \geq \Omega(r^2)$ and $\{Q_a\}$ are smoothed, if $\{\hat{Q}_a\}$ satisfy

$$\mathrm{Tr}(Q_a Q_b) = \mathrm{Tr}\left(\hat{Q}_a \hat{Q}_b\right) \text{ for all } a, b$$

$$\mathrm{Tr}(Q_a Q_b Q_c) = \mathrm{Tr}\left(\hat{Q}_a \hat{Q}_b \hat{Q}_c\right) \text{ for all } a, b, c$$

and $Q_1, \widehat{Q}_1$ are diagonal with sorted entries…

then $Q_a = \widehat{Q}_a$ for all a!

AN INEFFICIENT ALGORITHM

Input: $\{\mathrm{Tr}(Q_a Q_b)\}_{a,b}, \{\mathrm{Tr}(Q_a Q_b Q_c)\}_{a,b,c}$ (wlog Q_1 diagonal with sorted entries)

Consider the following **system of polynomial constraints**

Variables: $\hat{Q}_1, \dots, \hat{Q}_d$

Constraints:

For all $1 \leq a, b, c \leq d$:

- $\hat{Q}_a = \hat{Q}_a^\top$
- $\mathrm{Tr}(\hat{Q}_a \hat{Q}_b) = \mathrm{Tr}(Q_a Q_b)$
- $\mathrm{Tr}(\hat{Q}_a \hat{Q}_b \hat{Q}_c) = \mathrm{Tr}(Q_a Q_b Q_c)$
- $\hat{Q}_1$ is diagonal w/ sorted entries

The solution satisfies $\hat{Q}_a = Q_a$ for all a!

Polynomial system solving is NP-hard…

To get an efficient algorithm, we will design a suitable **convex relaxation** using the **sum-of-squares hierarchy**

SUM-OF-SQUARES: PROOFS TO ALGORITHMS

Powerful **generic** framework for algorithm design / nonconvex optimization

Inefficient algorithm with a **“simple”** proof of correctness → **Efficient** algorithm with the same guarantees

Yields the most powerful algorithms for many statistical problems

Robust regression, tensor decomposition, dictionary learning, matrix/tensor completion, sparse PCA, robust contextual bandits, Gaussian mixture models, differential privacy, robust mean estimation, community detection, clustering, robust Kalman filtering, robust structured distribution estimation…

SUM-OF-SQUARES: PROOFS TO ALGORITHMS

Idea 0: Instead of a single solution, find a **distribution** over solutions

…this is no easier than finding a single solution

SUM-OF-SQUARES: PROOFS TO ALGORITHMS

Idea 1: Find something which "behaves" like a distribution over solutions with respect to **low-degree test functions**

(Degree-t) Pseudo-expectation $\tilde{E}[\cdot]$:

Takes any degree-$\leq t$ polynomial in the variables $\hat{Q}_1, \ldots, \hat{Q}_d$ and outputs a number. Must satisfy:

1. **Normalization:** $\tilde{E}[1] = 1$
2. **Linearity:** $\tilde{E}[\alpha \cdot p + \beta \cdot q] = \alpha \cdot \tilde{\mathbb{E}}[p] + \beta \cdot \tilde{\mathbb{E}}[q]$
3. **Positivity:** $\tilde{E}[p^2] \geq 0$ for all degree-$\leq t/2$ polynomials p

The set of pseudo-expectations is convex, so we can **efficiently** find a **pseudo-distribution** over solutions to our polynomial system!

SUM-OF-SQUARES: PROOFS TO ALGORITHMS

Idea 2: If the proof of identifiability is **"simple"…**

i.e. every step involved a low-degree polynomial inequality like Cauchy-Schwarz

then because we proved that

$$\hat{Q}_a = Q_a \text{ for all } a$$

we conclude that $\tilde{E}[\hat{Q}_a] = Q_a$

(Degree-t) Pseudo-expectation $\tilde{E}[\cdot]$:

Takes any degree-$\leq t$ polynomial in the variables $\hat{Q}_1, \ldots, \hat{Q}_d$ and outputs a number. Must satisfy:

1. **Normalization:** $\tilde{E}[1] = 1$
2. **Linearity:** $\tilde{E}[\alpha \cdot p + \beta \cdot q] = \alpha \cdot \tilde{\mathbb{E}}[p] + \beta \cdot \tilde{\mathbb{E}}[q]$
3. **Positivity:** $\tilde{E}[p^2] \geq 0$ for all degree-$\leq t/2$ polynomials p

ALGORITHM

Input: $\{\mathrm{Tr}(Q_a Q_b)\}_{a,b}, \{\mathrm{Tr}(Q_a Q_b Q_c)\}_{a,b,c}$

1. Find pseudo-distribution $\tilde{E}$ over solutions to the polynomial system
2. Output $\tilde{E}[\hat{Q}_1], \ldots, \tilde{E}[\hat{Q}_d]$

Our proof of identifiability can be implemented in a simple fashion!

PRELIMINARIES

Motivation

Setup

Results

QUADRATIC CASE

Moments and tensor ring

Proof of identifiability

Algorithm

HIGHER-DEGREE CASE

TAKEAWAYS

MOMENTS

For higher-degree pushforwards, the moments are quite unwieldy…

We will only work with the second-order moments.

If $\mathcal{D}$ is a degree-ω pushforward given by $T_1, \dots, T_d \in (\mathbb{R}^r)^{\otimes \omega}$, then

$$\begin{aligned} \mathbb{E}_{z\sim\mathcal{D}}[z_a z_b] &= \mathbb{E}_{g\sim N(0,\mathrm{Id})}\left[\langle T_a, g^{\otimes\omega}\rangle\langle T_b, g^{\otimes\omega}\rangle\right] \\ &= \langle T_a, T_b\rangle_\Sigma \qquad \text{(in this talk, will pretend } \Sigma = \mathrm{Id}\text{)} \end{aligned}$$

Given inner products $\{\langle T_a, T_b\rangle\}$, can we recover $\{T_a\}$?

IDENTIFIABILITY

Given inner products $\{\langle T_a, T_b \rangle\}$, can we recover $\{T_a\}$?

If $\{T_a\}$ were arbitrary, this is just matrix factorization, in which case can only recover $\{T_a\}$ up to a rotation in $O(r^\omega)$, rather than $O(r)$

Thm [CLLZ]: $\{\langle T_a, T_b \rangle\}_{a,b}$ robustly determine $\{T_a\}$ up to gauge symmetry if $d \geq \Omega\left(r^{c\omega\ell}\right)$ and $\{T_a\}$ are rank-ℓ + smoothed.

i.e., **the degree-2 moments of a low-rank pushforward "robustly identify" the parameters**

PROOF OF IDENTIFIABILITY

Suppose $\{T_a\}$ and $\{\hat{T}_a\}$ are collections of rank-ℓ tensors satisfying

$$\langle T_a, T_b \rangle = \langle \hat{T}_a, \hat{T}_b \rangle \text{ for all } a, b$$

Want to show there exists $U \in O(r)$ for which

$$F_U(T_a) = \hat{T}_a \text{ for all } a = 1, \dots, d$$

PROOF OF IDENTIFIABILITY

Recall: $\{\langle T_a, T_b\rangle\}_{a,b}$ specifies $\{T_a\}$ **up to an $r^\omega \times r^\omega$ rotation**

i.e. exists $W \in O(r^\omega)$ for which $W \underbrace{\mathrm{vec}(T_a)}_{r^\omega \times 1} = \mathrm{vec}(\hat{T}_a)$

If we had $F_U(T_a) = \hat{T}_a$ for all a, then $W = U^{\otimes \omega}$

Any column $W^{i_1 \cdots i_\omega}$ of W, regarded as an $r \times \cdots \times r$ tensor, would be rank-1, specifically $W^{i_1 \cdots i_\omega} = U^{i_1} \otimes \cdots \otimes U^{i_\omega}$

Note: W maps many rank-ℓ tensors to other rank-ℓ tensors

We will use this to establish rank-1 structure of W

PROOF OF IDENTIFIABILITY

1. W vec(T) has rank-$\leq \ell$ whenever T is rank-ℓ

   because this is the case for many "incoherent" T

2. W vec(T) has rank-$\leq (\ell - 1)$ whenever T is rank-$(\ell - 1)$

   if this were not true for some rank-$(\ell - 1)$ T, then
   W vec$\left(T + v^{\otimes \omega}\right)$ would be rank $> \ell$ for some v, violating 1.

3. So W vec(T) has rank-1 whenever T is rank-1!

4. Take $T = e_{i_1} \otimes \cdots \otimes e_{i_\omega} \Rightarrow W$ vec(T) $=$ $\boldsymbol{W^{i_1 \cdots i_\omega}}$ **is rank-1**

Note:
- The above only applies to symmetric tensors T…
- Need to be careful when working with tensor rank…

PRELIMINARIES

Motivation

Setup

Results

QUADRATIC CASE

Moments and tensor ring

Proof of identifiability

Algorithm

HIGHER-DEGREE CASE

TAKEAWAYS

OPEN QUESTIONS

Is bounded rank necessary for parameter recovery when $\omega > 2$?

More practical algorithms?

Beyond polynomial activations?

Thm [C-Li-Li]: For pushforwards under depth-2 networks with **ReLU activations**, there is no polynomial-time algorithm even for outputting an arbitrary density which is $O(1)$-close in Wasserstein to the true distribution.

OPEN QUESTIONS

Is bounded rank necessary for parameter recovery when $\omega > 2$?

More practical algorithms?

Beyond polynomial activations?

Hardness of density estimation for polynomial activations?

New notions of distribution learning?

- e.g. learning in “computational” distance (“outcome indistinguishability”)

TAKEAWAYS

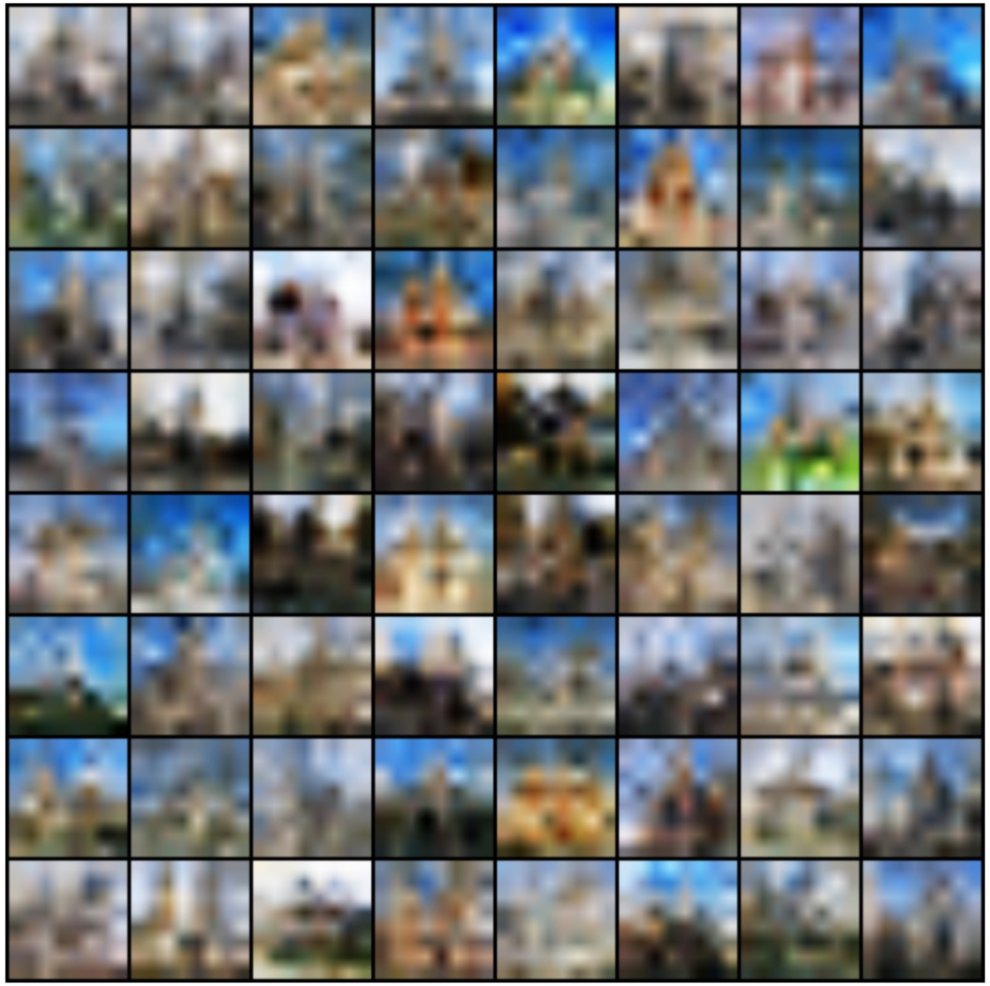
Pushforwards are a **powerful** way of modeling high-dimensional distributions **in practice**, but **very little known** w.r.t. **provable guarantees** / **principled ways of evaluating trained models**

Tools / perspectives from **TCS** well-suited to fill this gap

Pseudorandom generators, distribution learning theory, smoothed analysis, method of moments / tensor methods, convex programming hierarchies

By building on these ideas, we give the **first efficient algorithms** for **provably learning** a natural family of pushforward distributions

Lots of open questions! (both technical and conceptual)




MOLDOVITA MONAST