Efficient Learning of Generative Models via Finite-Difference Score Matching

Tianyu Pang*[1], Kun Xu*[1], Chongxuan Li[1], Yang Song[2], Stefano Ermon[2], and Jun Zhu[1]

[1]Department of Computer Science and Technology, Tsinghua University

[2]Department of Computer Science, Stanford University

NEURAL INFORMATION PROCESSING SYSTEMS

Introduction

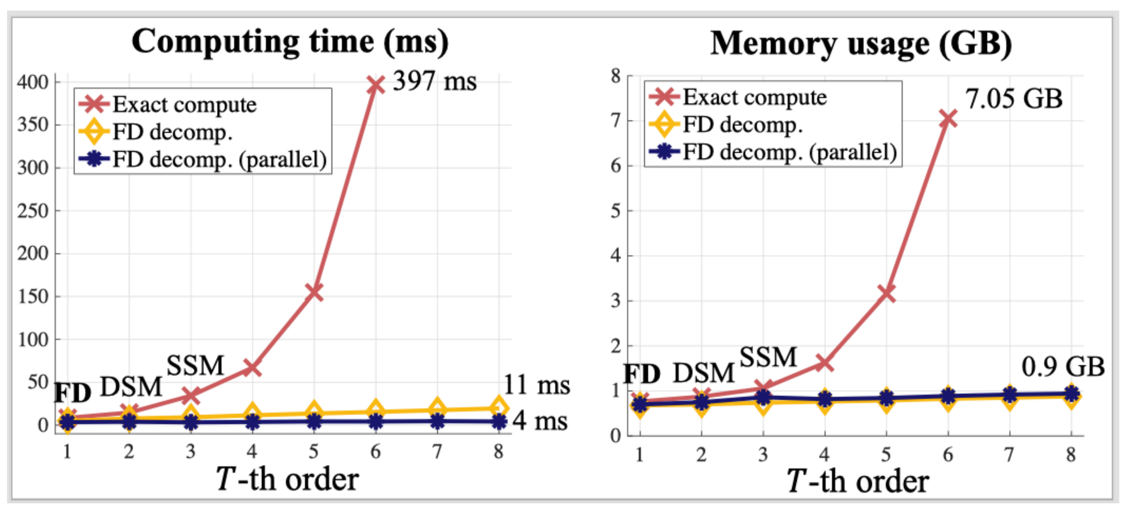
Energy-based model is flexible:

$$p(x) = \frac{\exp(-E(x))}{Z}, Z = \int \exp(-E(x))dx,$$

And can be trained by Score Matching (SM):

$$\mathcal{J}_{SM}(\theta) = E_{p_D(x)}\Big[\frac{1}{2}\underbrace{||\nabla_x \log p(x)||_2^2}_{\text{1st order derivatives}} + tr(\underbrace{\nabla_x^2 \log p(x)}_{\text{2nd order derivatives}})\Big].$$

However, higher order derivatives in deep learning is computational expensive:



We propose using Finite Difference (FD) to approximate higher derivatives.

Methodology

There existing a set of coefficients $\{\gamma_i\}_{i=1:T+1}$ and $\{\beta_i\}_{i=1:T+1}$, such that:

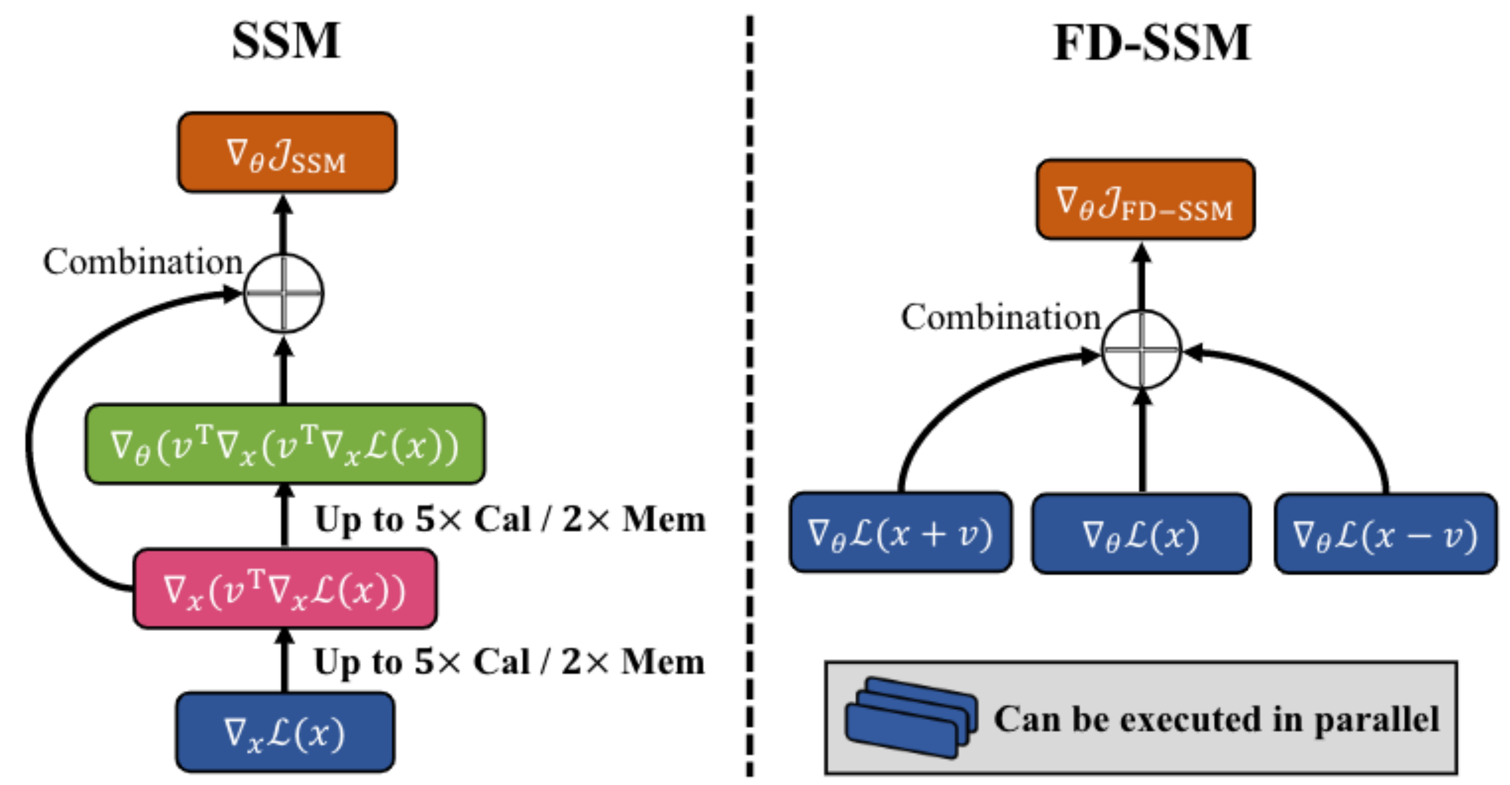
$$\frac{\partial^T}{\partial v^T}\mathcal{L}_\theta(x) = \frac{T!}{\epsilon^T}\sum_{i=1}^{T+1}\beta_i\mathcal{L}_\theta(x+\gamma_i v) + o(1).$$

The 1st order and 2nd order derivatives is given as:

$$\begin{cases} v^\top\nabla_x\mathcal{L}_\theta(x) = \epsilon\frac{\partial}{\partial v}\mathcal{L}_\theta(x) = \frac{1}{2}\mathcal{L}_\theta(x+v) - \frac{1}{2}\mathcal{L}_\theta(x-v) + o(\epsilon^2); \\ v^\top\nabla_x^2\mathcal{L}_\theta(x)v = \epsilon^2\frac{\partial^2}{\partial v^2}\mathcal{L}_\theta(x) = \mathcal{L}_\theta(x+v) + \mathcal{L}_\theta(x-v) - 2\mathcal{L}_\theta(x) + o(\epsilon^3). \end{cases}$$

And the corresponding FD-SSM objective function is:

$$\mathcal{J}_{\text{FD-SSM}}(\theta) = \frac{1}{\epsilon^2}\mathbb{E}_{p_{\text{data}}(x)}\mathbb{E}_{p_\epsilon(v)}\Big[\log p_\theta(x+v) + \log p_\theta(x-v) - 2\log p_\theta(x) + \frac{1}{8}\left(\log p_\theta(x+v) - \log p_\theta(x-v)\right)^2\Big] = \mathcal{J}_{\text{SSM}}(\theta) + o(\epsilon).$$



Our contributions:

- An approximation of higher order derivatives with linear time and memory complexity
- With theoretical guaranteed consistency
- Accelerate score matching method

Experimental Results

FD-SM on deep EBM: Out-of-distribution Detection

Dataset	Algorithm	Time	SVHN	CIFAR	ImageNet
SVHN	DSM	673 ms	0.49	1.00	0.99
	FD-DSM	**305 ms**	0.50	1.00	1.00
CIFAR	DSM	635 ms	0.91	0.49	0.79
	FD-DSM	**311 ms**	0.92	0.51	0.81
ImageNet	DSM	1125 ms	0.95	0.87	0.49
	FD-DSM	**713 ms**	0.95	0.89	0.49

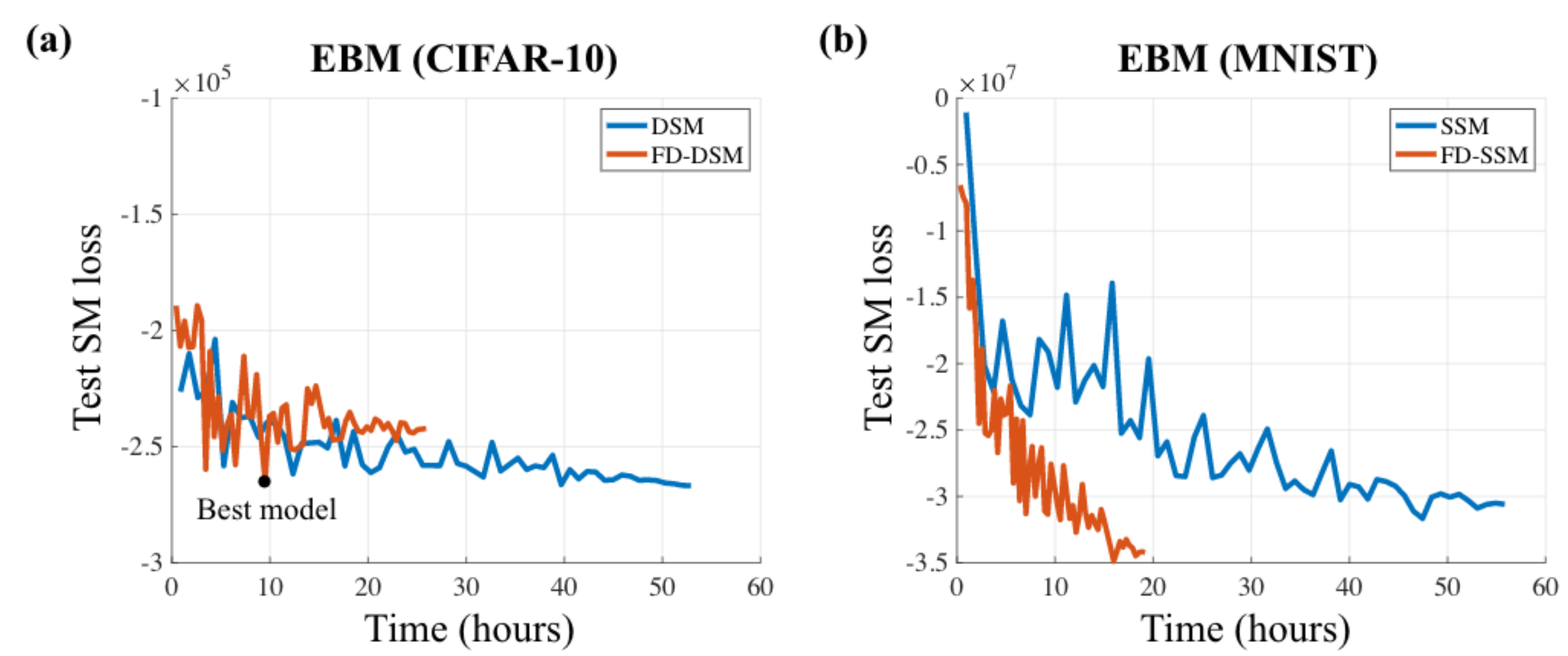
Deep EBM on MNIST: Consistent performance with better time & memory usage

Algorithm	SM loss	Time	Mem.
DSM	-9.47×10^4	282 ms	3.0 G
FD-DSM*	-9.24×10^4	191 ms	3.2 G
FD-DSM	-9.27×10^4	**162 ms**	2.7 G
SSM	-2.97×10^7	673 ms	5.1 G
SSMVR	-3.09×10^7	670 ms	5.0 G
FD-SSM*	-3.36×10^7	276 ms	3.7 G
FD-SSM	-3.33×10^7	**230 ms**	3.4 G

Generated results of deep EBM

MNIST | Fashion-MNIST | CelebA

Learning curve of FD-SM vs. SM



Primary Contacts: Tianyu Pang (pty17@mails.tsinghua.edu.cn), Kun Xu (kunxu.thu@gmail.com); Code: https://github.com/taufikxu/FD-ScoreMatching