NEURAL INFORMATION PROCESSING SYSTEMS

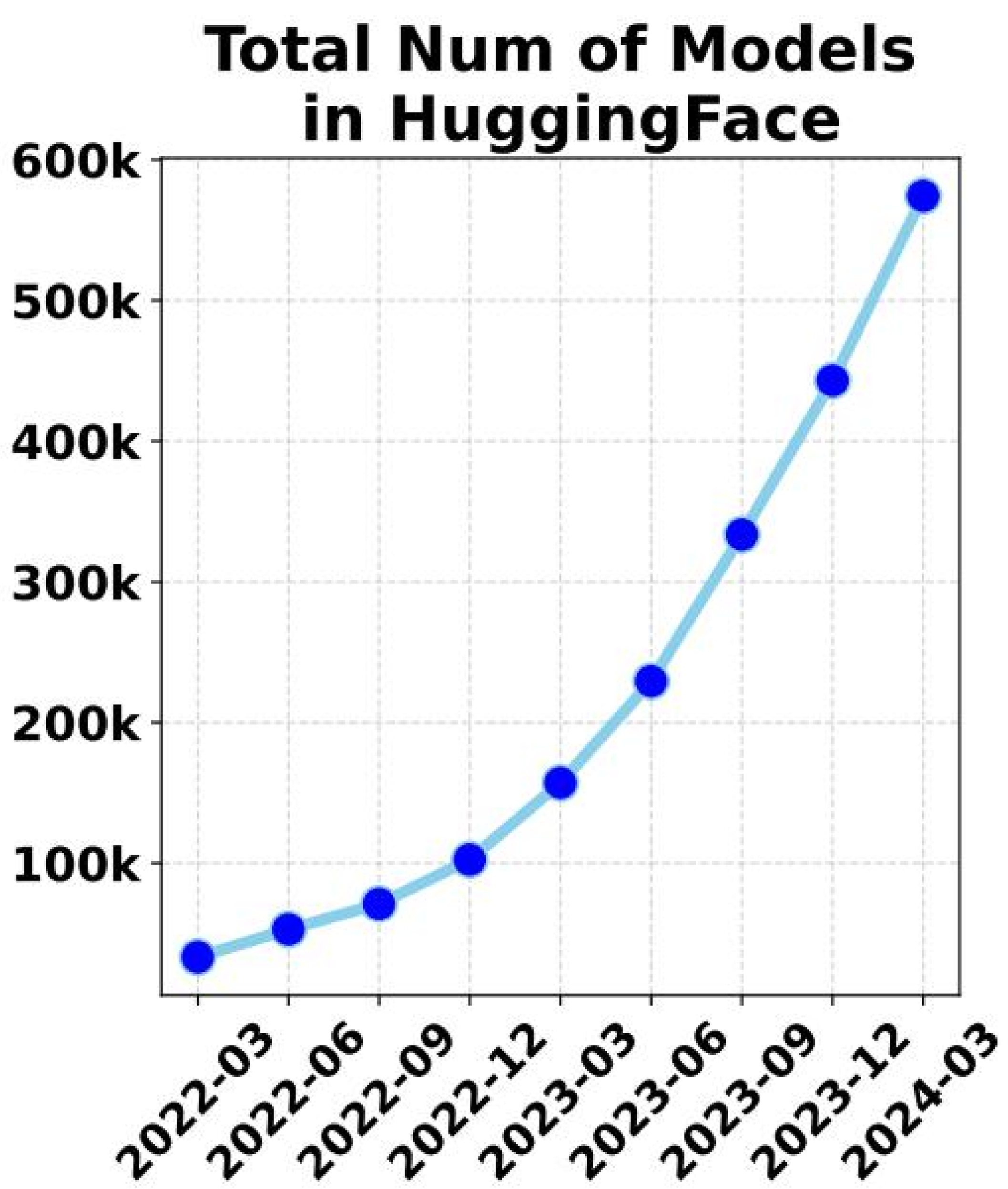
FM-Delta: Lossless Compression for Storing Massive Fine-tuned Foundation Models

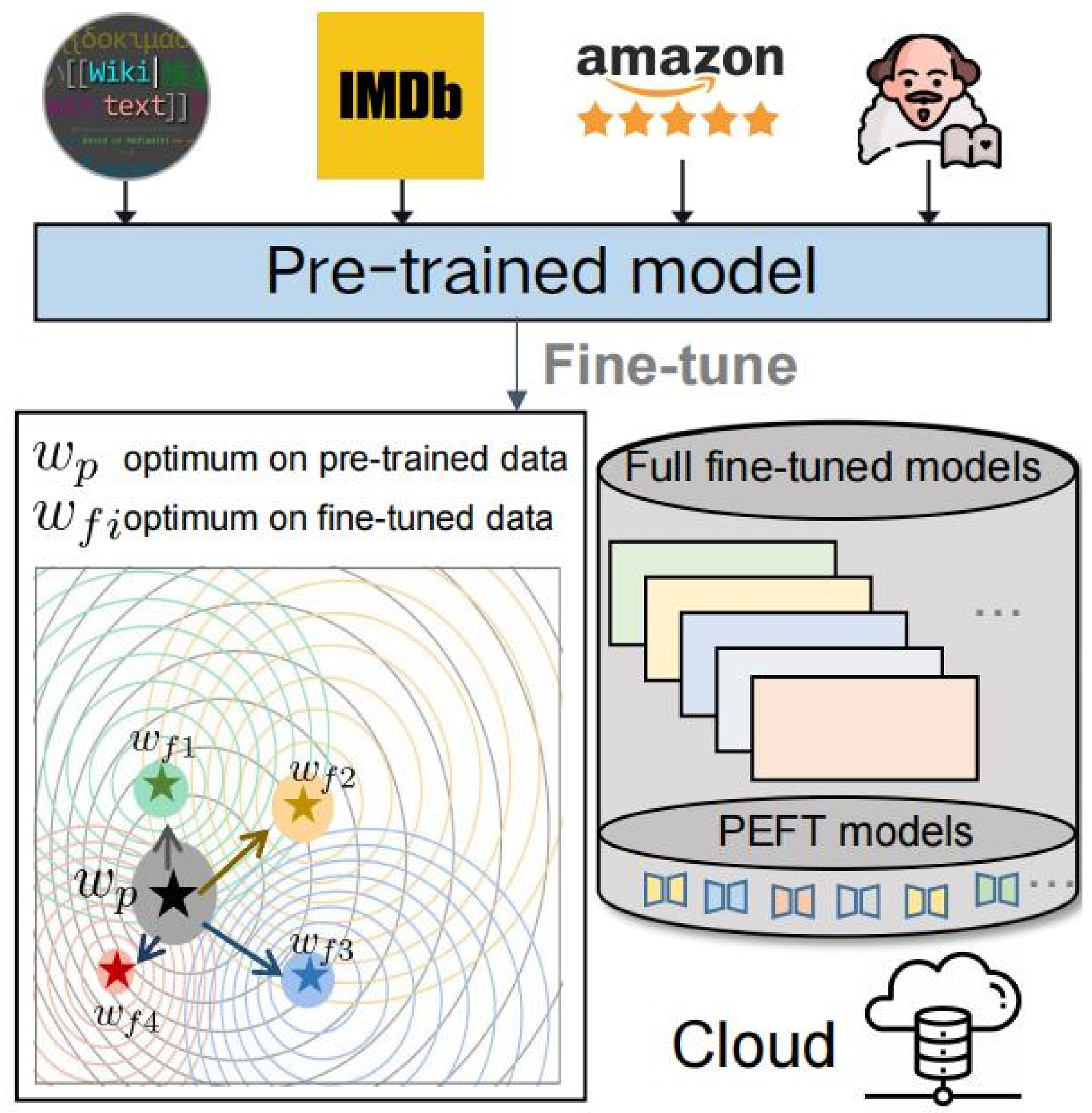
Wanyi Ning[1], Jingyu Wang[1], Qi Qi[1], Mengde Zhu[1], Haifeng Sun[1], Daixuan Cheng[1], Jianxin Liao[1], Ce Zhang[2]

[1]Beijing University of Posts and Telecommunications [2]University of Chicago

北京郵電大學
Beijing University of Posts and Telecommunications

Motivation: Storage Overhead in Cloud





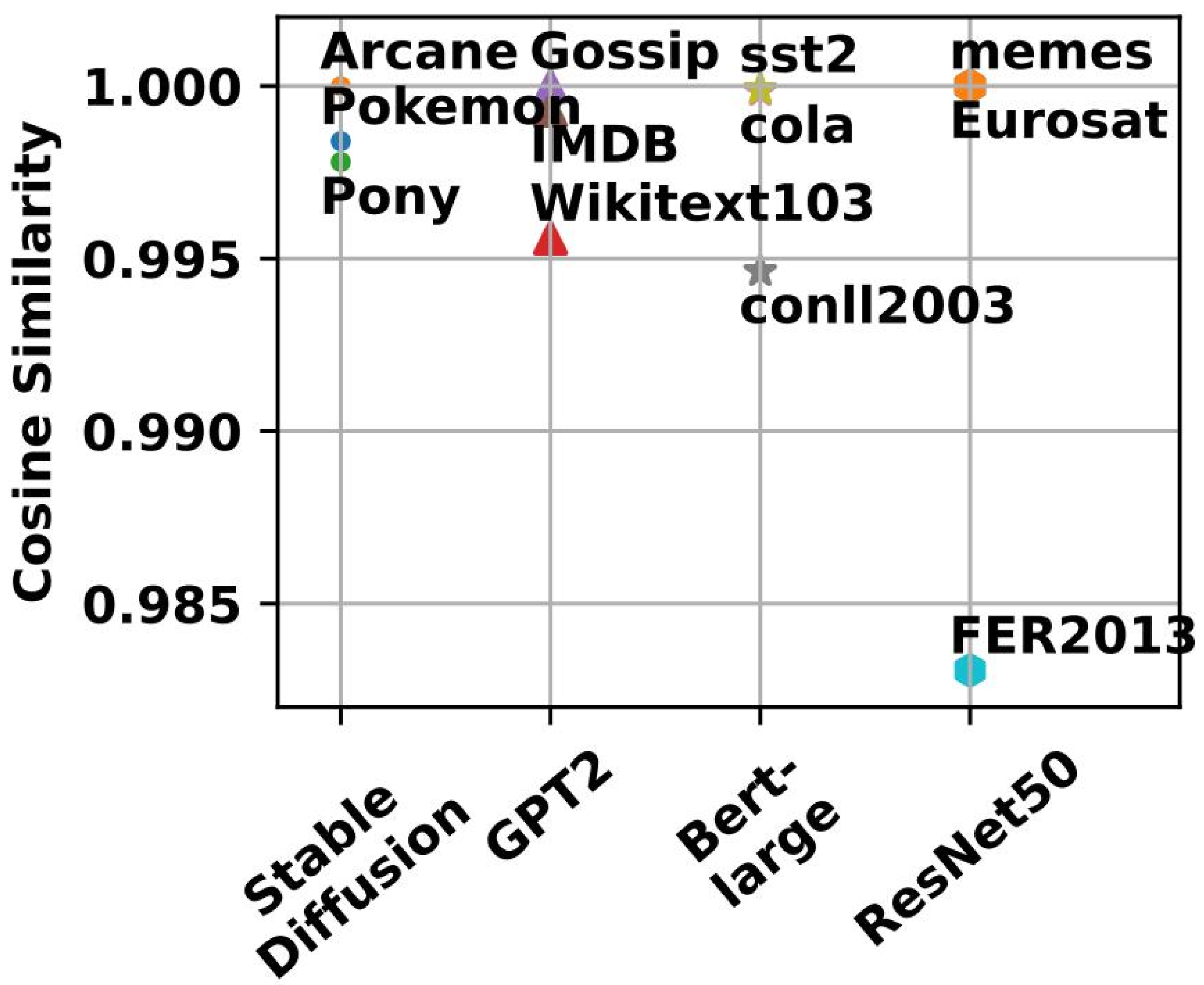
Model	Model size	Full num.	PEFT num.	Inact.
Falcon-40B	40B	79	48	82%
GPT-NeoX	20B	51	22	84%
GPT-J	6B	284	75	88%
LLaMA-7B	7B	5112	1170	91%
Bert-large	336M	260	159	88%
Stable Diff.	860M	1606	65	64%
Approx. disk storage		159TB	4TB	89%

$9,540

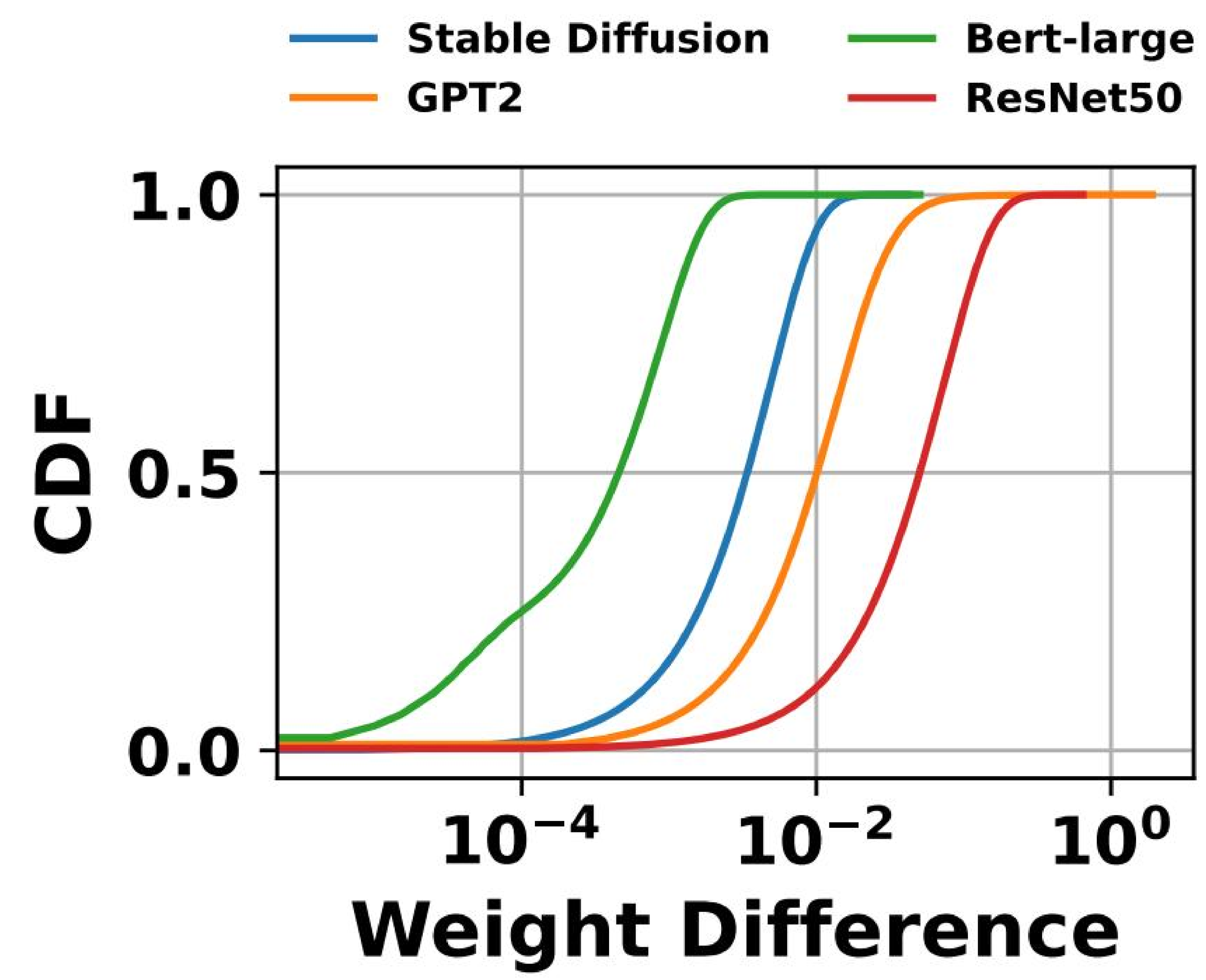
- ***heavy storage overhead*** for cloud providers such as HuggingFace.
- key pain point --- **full fine-tuned models**.
- premise --- protect users' intellectual property of users (i.e., **no changes to models**)

Finding: Small Delta

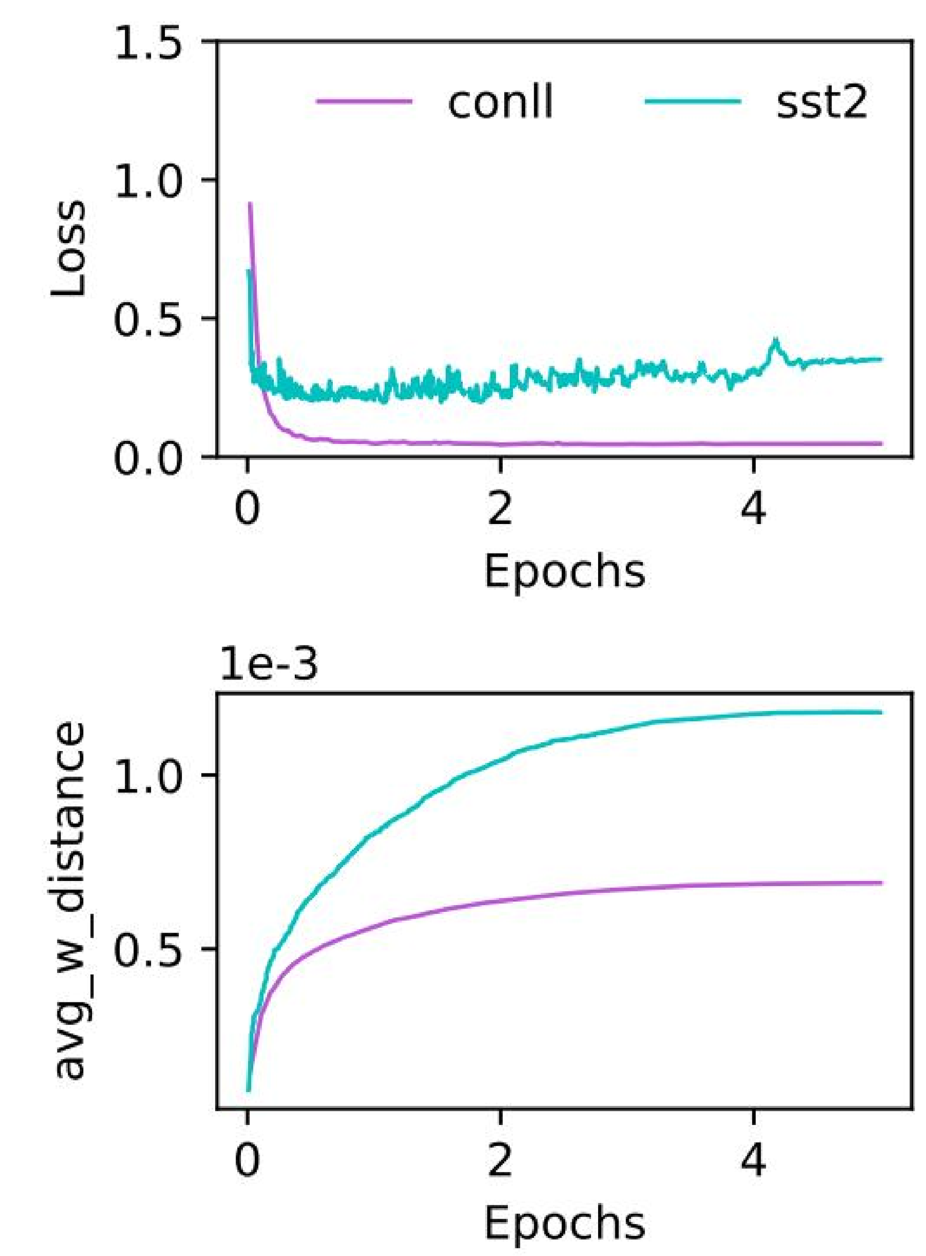
- Empirical analysis: **a small difference (delta)** *between most* fine-tuned and pre-trained models stored in cloud.



- High model cosine similarity



- Small element difference 10^{-4}-10^{0}



(b) Bert-large-uncased

- Average difference grows slow over fine-tuning.

Finding: Small Delta

- Theoretical analysis: delta grows slowly as the number of fine-tuning steps T increases.

Assumption 1. For the loss function f, there exists $\mathbf{w}^* \in \mathbb{R}^d$ such that $f(\mathbf{w}) \geq f(\mathbf{w}^*)$, for all $\mathbf{w}$.

Assumption 2. f satisfies that for all $\mathbf{w}, \mathbf{v} \in \mathbb{R}^d$, $f(\mathbf{w}) - f(\mathbf{v}) \leq (\mathbf{w} - \mathbf{v})^T \nabla f(\mathbf{v}) + \frac{\beta}{2}\|\mathbf{w} - \mathbf{v}\|^2$.

Assumption 3. Given a data distribution $\mathcal{D}$, the variance of stochastic gradient is bounded: $\mathbb{E}_{\xi\sim\mathcal{D}} \|G(\mathbf{w};\xi) - \nabla f(\mathbf{w})\|^2 \leq \sigma^2$.

Theorem 1 (Growth Rate for Model Difference.). *Let $\mathbf{w}_p$ and $\mathbf{w}_f$ are the parameters of the pre-trained and fine-tuned models, respectively. The fine-tuning stage involves T training steps. With learning rate $\eta_t = \frac{1}{\beta\sqrt{t}}$, $t = 1, 2, ..., T$, the distance between $\mathbf{w}_p$ and $\mathbf{w}_f$ is*

$$\mathbb{E}\left[||\mathbf{w}_f - \mathbf{w}_p||\right] \leq \frac{\sqrt{3}\sigma}{\beta} + C_1(\ln T)^{\frac{1}{2}} + \boxed{C_2 T^{\frac{1}{4}}}. \tag{2}$$

where $||\cdot||$ is l_2-norm; f is the β-smooth convex loss function on the fine-tuning dataset; $\mathbf{w}^*$ is the optimal model parameter on the fine-tuning task; C_1 and C_2 are the constants related to the pre-trained model, which are $C_1 = \left(\frac{9\sigma^2}{4\beta^2} + \frac{f(\mathbf{w}_p)-f(\mathbf{w}^*)}{2\beta}\right)^{\frac{1}{2}}$ and $C_2 = \left(\frac{\sigma^2}{\beta^2} + \frac{2(f(\mathbf{w}_p)-f(\mathbf{w}^*))}{\beta}\right)^{\frac{1}{2}}$

Method: FM-Delta

Step 1. Mapping Float into Integer for Delta:

- map floating-points into *unsigned integers*, and performs *integer subtraction*.

		byte number	1 2 3 4
w_f	0.0316	int(w_p)	3d 01 6f 00
w_p	0.0309	int(w_f)	3c fd 21 ff
$w_f - w_p$	0.0007	int($w_f - w_p$)	3a 37 80 34
		int(w_f)-int(w_p)	00 04 4d 01

Method: FM-Delta

Step 2. Compression with Range Coding:

- Symbolization: <sign ***s***, most significant bit ***k***> of int delta as symbols.
- Probability Model: a quasi-static probability modeler to termly update symbol frequencies.
- Encoding: the symbols with the raw bits on all delta elements through range scaling.
- Decoding: map the encoded value back to the original symbol range --> reverse-mapping delta --> original float fine-tuned model

Results: Compression Rate

Family	Pretrained Size	Finetuned Num.	Original Storage (GB)	Storage after Compression (GB)					
				LZMA	Gzip	Zlib	FPZip	BZip2	FM-Delta
Falcon-40B (fp16)	40B	5	461.6	349.3	373.4	373.4	456.9	342.7	**270.8 (59%)**
		10	846.3	621.7	669.9	669.9	837.8	608.5	**473.9 (56%)**
GPT-NeoX (fp16)	20B	5	230.8	162.9	177.2	176.4	213.4	158.6	**112.4 (49%)**
		10	423.2	298.7	324.9	323.4	391.2	290.7	**205.2 (48%)**
GPT-J (fp16)	6B	5	68.4	57.2	60.6	60.6	61.2	58.7	**44.6 (65%)**
		10	125.3	104.8	111	111	112.2	107.6	**73.8 (59%)**
GPT-2	124M	50	24.2	21.8	22	22	21.9	22.5	**15 (62%)**
		100	48	43.2	43.5	43.5	43.4	44.5	**28.7 (60%)**
Bert-large-uncased	336M	50	63.7	58.6	59.1	59.1	58.9	60.4	**41.3 (65%)**
		100	126.1	116.1	117.1	117.1	116.6	119.6	**82.1 (65%)**
Stable-Diffusion UNet	860M	5	19.2	17.7	17.8	17.8	17.8	18.3	**12.8 (67%)**
		10	35.2	32.5	32.7	32.7	32.6	33.5	**23.5 (67%)**
ResNet50	26M	10	1.1	0.9	0.9	0.9	0.9	0.9	**0.7 (68%)**
		20	2	1.7	1.7	1.7	1.7	1.8	**1.3 (66%)**
Avg. Compression Throughput (MB/s)				4.9	36.1	35.6	83.5	12.1	**109.7**
Avg. Decompression Throughput (MB/s)				24.8	236.6	**260.8**	80.6	23.8	100.9

Results: Cloud Cost Analysis

Is the cost of decompressing models lower than storing them uncompressed in the cloud?

- regard model download as a binomial distribution

$$P(X = k) = C_n^k \cdot \left(\frac{10}{30 \times 24 \times 60}\right)^k \cdot \left(1 - \frac{10}{30 \times 24 \times 60}\right)^{n-k}$$

k represents the number of concurrent download requests in a given minute.

Goal: maximize loadable models *n*, s.t. $\sum_{k=t}^{n} P(X = k) \leq 0.01$

⇨ $n_{\max} = 35300$

100% storage vs. **50% storage + server purchase fee** ⟶ ***40% cost reduction***

Thank you!

Feel free to contact me: ningwanyi@bupt.edu.cn