Neural Networks for Learnable and Scalable Influence Estimation of Instruction Fine-Tuning Data

Ishika Agarwal, Dilek Hakkani-Tür

NEURAL INFORMATION PROCESSING SYSTEMS

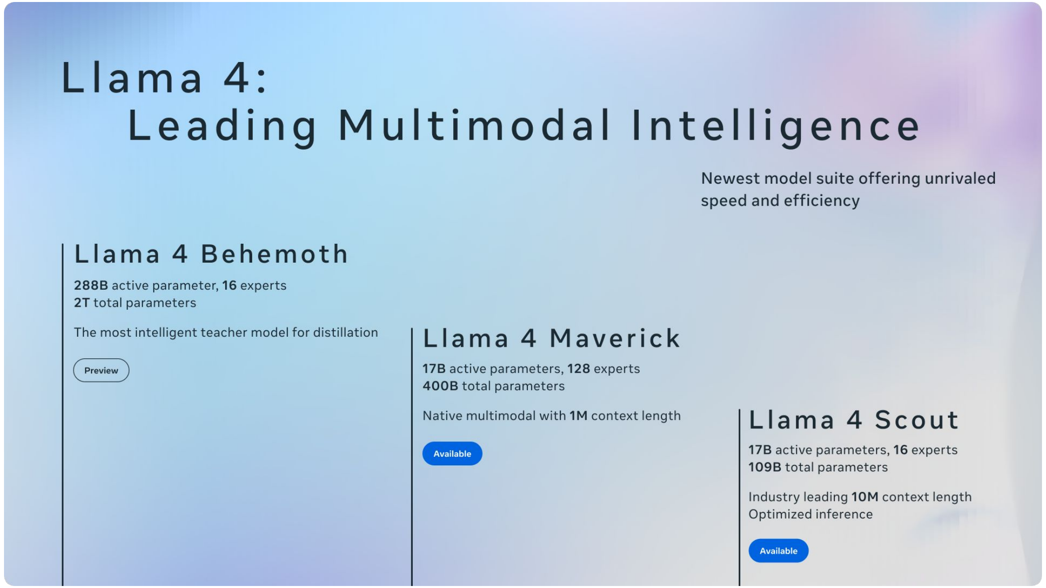
UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN

CONVAI

CDS Siebel School of Computing and Data Science

LLM training is expensive

- Llama 4
- Rumored to have been pre-trained on 32k H100's
- Quality data is scarce



Solutions

1. More efficient model architectures
2. Faster hardware
3. Fewer data

Efficient Memory Management for Large Language Model Serving with *PagedAttention*

Woosuk Kwon[1,*] Zhuohan Li[1,*] Siyuan Zhuang[1] Ying Sheng[1,2] Lianmin Zheng[1] Cody Hao Yu[3] Joseph E. Gonzalez[1] Hao Zhang[4] Ion Stoica[1]

[1]UC Berkeley [2]Stanford University [3]Independent Researcher [4]UC San Diego

FlashAttention: Fast and Memory-Efficient Exact Attention with IO-Awareness

Tri Dao[†], Daniel Y. Fu[†], Stefano Ermon[†], Atri Rudra[‡], and Christopher Ré[†]

[†]Department of Computer Science, Stanford University
[‡]Department of Computer Science and Engineering, University at Buffalo, SUNY

{trid,danfu}@cs.stanford.edu, ermon@stanford.edu, atri@buffalo.edu, chrismre@cs.stanford.edu

Solutions

1. More efficient model architectures
2. Faster hardware
3. Fewer data

In-Datacenter Performance Analysis of a Tensor Processing Unit™

Norman P. Jouppi, Cliff Young, Nishant Patil, David Patterson, Gaurav Agrawal, Raminder Bajwa, Sarah Bates, Suresh Bhatia, Nan Boden, Al Borchers, Rick Boyle, Pierre-luc Cantin, Clifford Chao, Chris Clark, Jeremy Coriell, Mike Daley, Matt Dau, Jeffrey Dean, Ben Gelb, Tara Vazir Ghaemmaghami, Rajendra Gottipati, William Gulland, Robert Hagmann, C. Richard Ho, Doug Hogberg, John Hu, Robert Hundt, Dan Hurt, Julian Ibarz, Aaron Jaffey, Alek Jaworski, Alexander Kaplan, Harshit Khaitan, Daniel Killebrew, Andy Koch, Naveen Kumar, Steve Lacy, James Laudon, James Law, Diemthu Le, Chris Leary, Zhuyuan Liu, Kyle Lucke, Alan Lundin, Gordon MacKean, Adriana Maggiore, Maire Mahony, Kieran Miller, Rahul Nagarajan, Ravi Narayanaswami, Ray Ni, Kathy Nix, Thomas Norrie, Mark Omernick, Narayana Penukonda, Andy Phelps, Jonathan Ross, Matt Ross, Amir Salek, Emad Samadiani, Chris Severn, Gregory Sizikov, Matthew Snelham, Jed Souter, Dan Steinberg, Andy Swing, Mercedes Tan, Gregory Thorson, Bo Tian, Horia Toma, Erick Tuttle, Vijay Vasudevan, Richard Walter, Walter Wang, Eric Wilcox, and Doe Hyun Yoon

Google, Inc., Mountain View, CA USA

Email: {jouppi, cliffy, nishantpatil, davidpatterson} @google.com

Training Giant Neural Networks Using Weight Streaming on Cerebras Wafer-Scale Clusters

Stewart Hall, Rob Schreiber, Sean Lie, Cerebras Systems, Inc.

Solutions

1. More efficient model architectures
2. Faster hardware
3. Fewer data

LESS: Selecting Influential Data for Targeted Instruction Tuning

Mengzhou Xia [1 *] Sadhika Malladi [1 *] Suchin Gururangan [2] Sanjeev Arora [1] Danqi Chen [1]

VisualGPT: Data-efficient Adaptation of Pretrained Language Models for Image Captioning

Jun Chen[1] , Han Guo[2], Kai Yi [1], Boyang Li[3], Mohamed Elhoseiny[1]
[1] King Abdullah University of Science and Technology (KAUST),
[2]Carnegie Mellon University, [3] Nanyang Technological University
{jun.chen,kai.yi,mohamed.elhoseiny}@kaust.edu.sa
hanguo@cs.cmu.edu, boyang.li@ntu.edu.sg

Solutions

1. More efficient model architectures
2. Faster hardware
3. Fewer data

LESS: Selecting Influential Data for Targeted Instruction Tuning

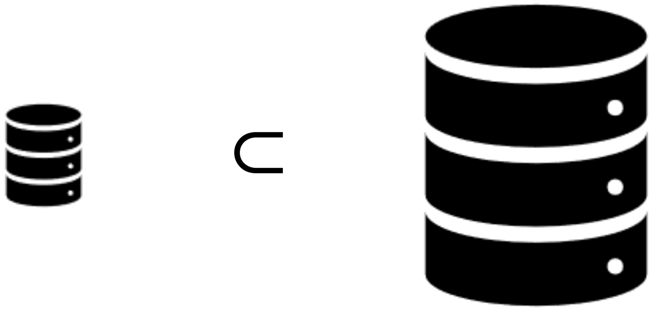
Mengzhou Xia [1 *] **Sadhika Malladi** [1 *] **Suchin Gururangan** [2] **Sanjeev Arora** [1] **Danqi Chen** [1]

VisualGPT: Data-efficient Adaptation of Pretrained Language Models for Image Captioning

Jun Chen[1] , Han Guo[2], Kai Yi [1], Boyang Li[3], Mohamed Elhoseiny[1]
[1] King Abdullah University of Science and Technology (KAUST),
[2]Carnegie Mellon University, [3] Nanyang Technological University
{jun.chen,kai.yi,mohamed.elhoseiny}@kaust.edu.sa
hanguo@cs.cmu.edu, boyang.li@ntu.edu.sg

Influence estimation…?

- How important is this data point?
- i.e. influence function (Koh & Liang 2017) or data valuation
- Use it to choose a subset of influential samples



How is influence estimation done?

- Model Dependent

- Model Independent

How is influence estimation done?

- Model Dependent
  - Subset is specific to model’s weaknesses
  - Uses model-specific signals like confidence, performance, gradients

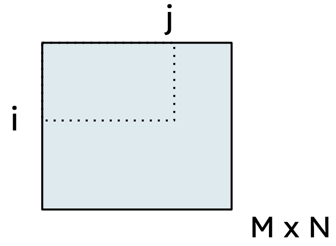
- Model Independent
  - Subset can be used for any model
  - Uses clustering-based or semantic similarity based methods

How is influence estimation done?

- Model Dependent
  - Subset is specific to model’s weaknesses
  - Uses model-specific signals like confidence, performance, gradients

- Model Independent
  - Subset can be used for any model
  - Uses clustering-based or semantic similarity based methods

- sim(i, j) = Influence of j on i



Influence estimation is expensive!

- Forward/backprop using a language model

Method	Cost	Size
Pairwise		
DELIFT (Agarwal et al., 2025)	$\mathcal{O}(MN) \cdot F$	7-8B
DELIFT (SE) (Agarwal et al., 2025)	$\mathcal{O}(MN) \cdot F$	355M
LESS (Xia et al., 2024)	$\mathcal{O}(M+N) \cdot B$	7-8B
(spoiler)		
Pointwise		
SelectIT (Liu et al., 2024a)	$\mathcal{O}(M) \cdot F$	7-8B
(spoiler)		

Motivation

- Inference/backprop using a language model

Method	Cost	Size
Pairwise		
DELIFT (Agarwal et al., 2025)	$\mathcal{O}(MN) \cdot F$	7-8B
DELIFT (SE) (Agarwal et al., 2025)	$\mathcal{O}(MN) \cdot F$	355M
LESS (Xia et al., 2024)	$\mathcal{O}(M+N) \cdot B$	7-8B
(spoiler)		
Pointwise		
SelectIT (Liu et al., 2024a)	$\mathcal{O}(M) \cdot F$	7-8B
(spoiler)		

These works estimate influence but…

Motivation

- Inference/backprop using a language model

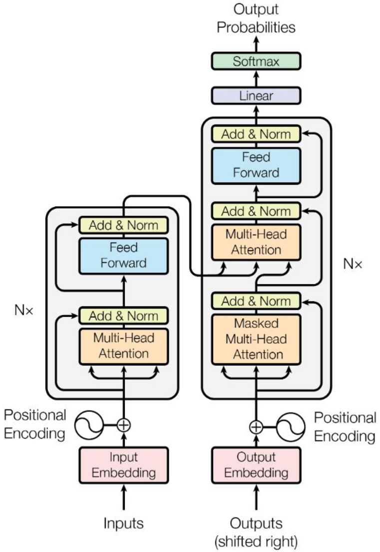
Method	Cost	Size
Pairwise		
DELIFT (Agarwal et al., 2025)	$\mathcal{O}(MN) \cdot F$	7-8B
DELIFT (SE) (Agarwal et al., 2025)	$\mathcal{O}(MN) \cdot F$	355M
LESS (Xia et al., 2024)	$\mathcal{O}(M+N) \cdot B$	7-8B
(spoiler)		
Pointwise		
SelectIT (Liu et al., 2024a)	$\mathcal{O}(M) \cdot F$	7-8B
(spoiler)		

Can we *learn to estimate* influence instead?

Motivation

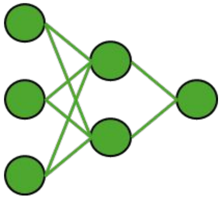
LLM-based influence functions

- Input: text data
- Output: influence score



Train a neural network:

- Input: (embedded) text data
- Output: influence score



Motivation

Method	Cost	Size
Pairwise		
DELIFT (Agarwal et al., 2025)	$\mathcal{O}(MN) \cdot F$	7-8B
DELIFT (SE) (Agarwal et al., 2025)	$\mathcal{O}(MN) \cdot F$	355M
LESS (Xia et al., 2024)	$\mathcal{O}(M+N) \cdot B$	7-8B
(spoiler)		
Pointwise		
SelectIT (Liu et al., 2024a)	$\mathcal{O}(M) \cdot F$	7-8B
(spoiler)		

Motivation

Method	Cost	Size
Pairwise		
DELIFT (Agarwal et al., 2025)	$\mathcal{O}(MN) \cdot F$	7-8B
DELIFT (SE) (Agarwal et al., 2025)	$\mathcal{O}(MN) \cdot F$	355M
LESS (Xia et al., 2024)	$\mathcal{O}(M+N) \cdot B$	7-8B
NN-CIFT (ours)	$\mathcal{O}(MN) \cdot F$	205K
Pointwise		
SelectIT (Liu et al., 2024a)	$\mathcal{O}(M) \cdot F$	7-8B
NN-CIFT (ours)	$\mathcal{O}(M) \cdot F$	205K

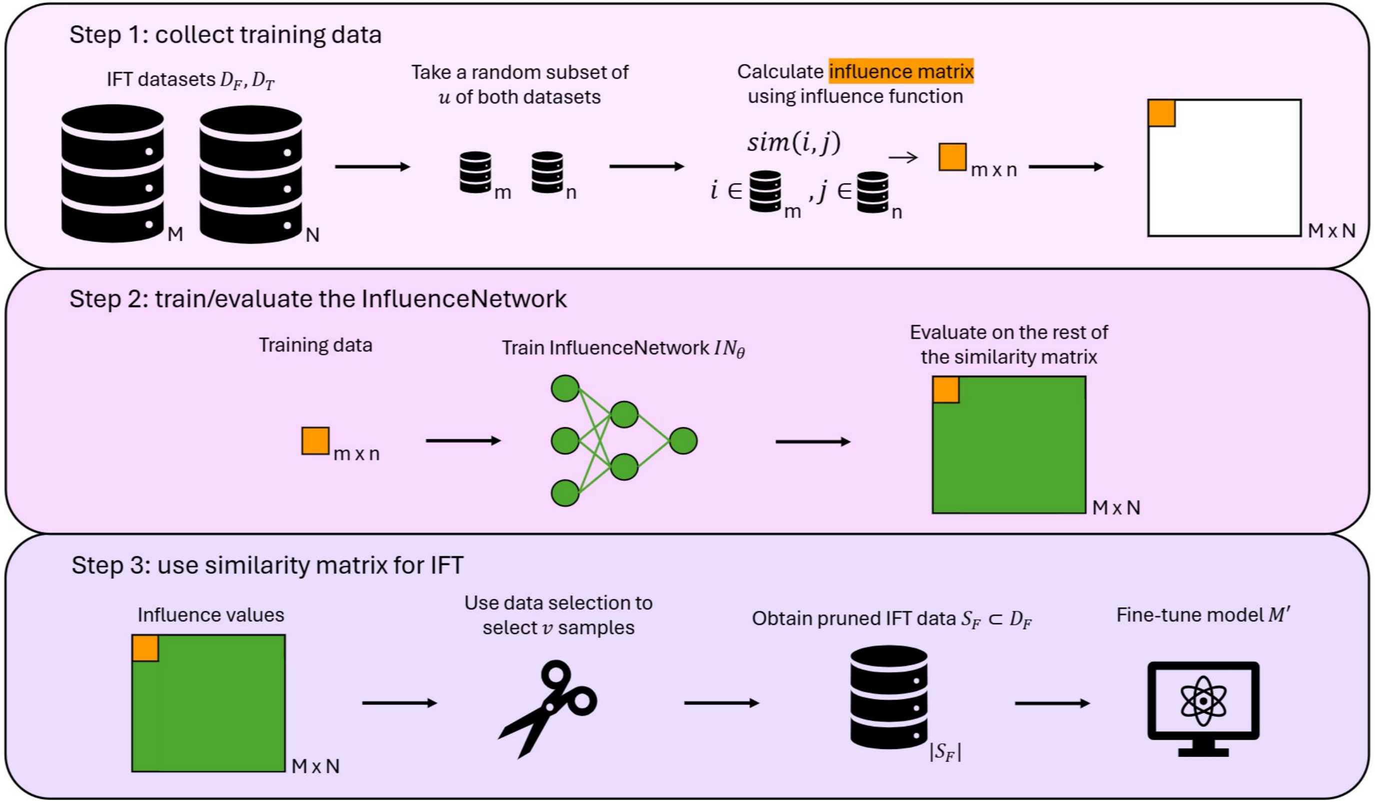
Motivation

CONVAI

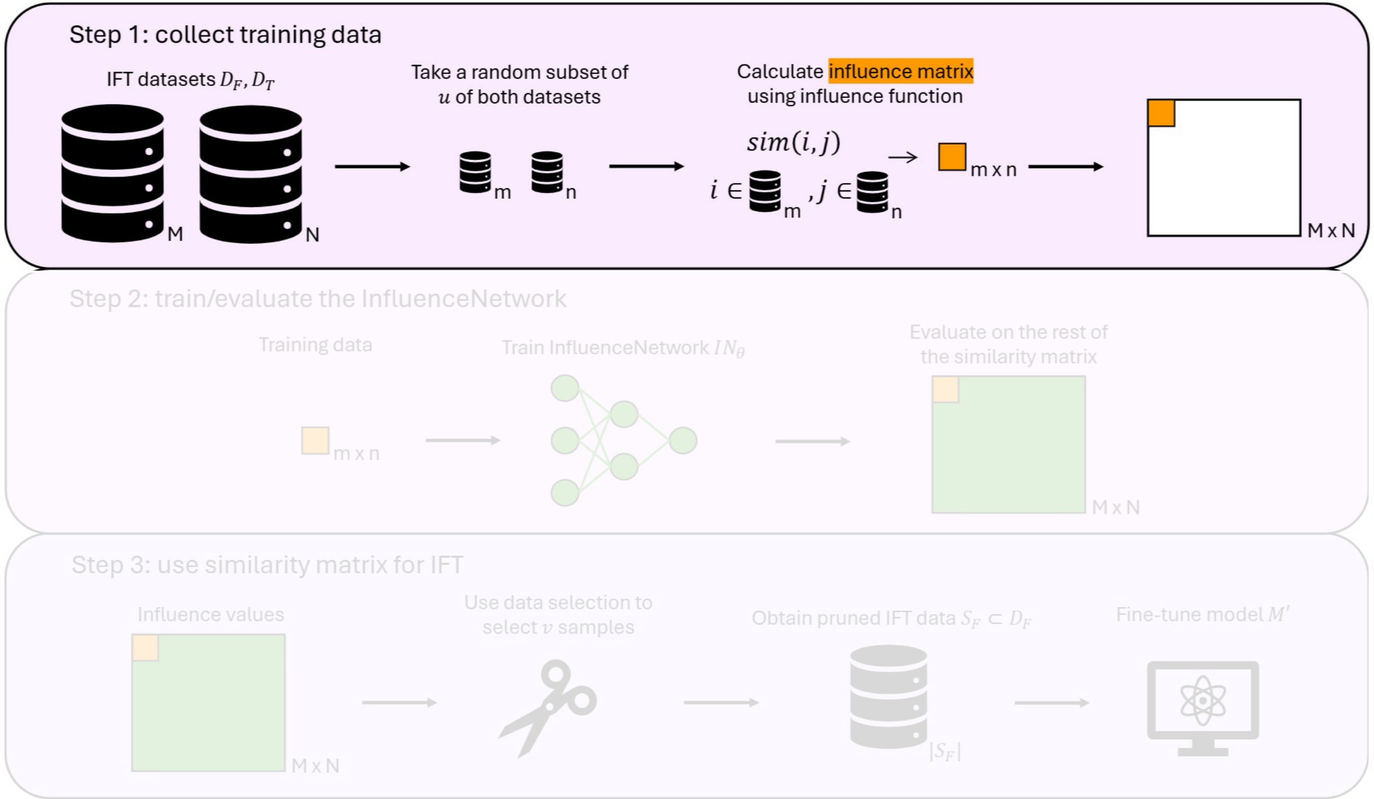
Method	Cost	Size
Pairwise		
DELIFT (Agarwal et al., 2025)	$\mathcal{O}(MN) \cdot F$	7-8B
DELIFT (SE) (Agarwal et al., 2025)	$\mathcal{O}(MN) \cdot F$	355M
LESS (Xia et al., 2024)	$\mathcal{O}(M+N) \cdot B$	7-8B
NN-CIFT (ours)	$\mathcal{O}(MN) \cdot F$	205K
Pointwise		
SelectIT (Liu et al., 2024a)	$\mathcal{O}(M) \cdot F$	7-8B
NN-CIFT (ours)	$\mathcal{O}(M) \cdot F$	205K

Model size reduces by 99.73%!

NN-CIFT: Neural Networks for Efficient Instruction Fine-Tuning



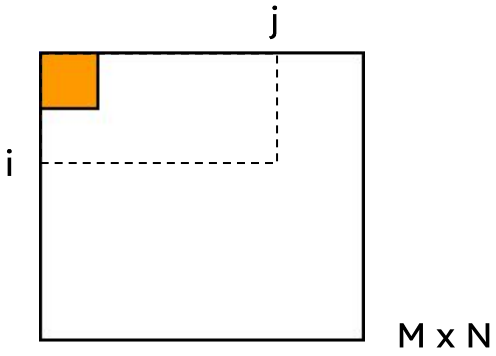
NN-CIFT



Step 1: collect training data
IFT datasets D_F, D_T
M
N
Take a random subset of u of both datasets
m
n
Calculate influence matrix using influence function
$sim(i,j)$
$i \in$ m $, j \in$ n
$\rightarrow$ m x n
M x N
Step 2: train/evaluate the InfluenceNetwork
Training data
m x n
Train InfluenceNetwork IN_θ
Evaluate on the rest of the similarity matrix
M x N
Step 3: use similarity matrix for IFT
Influence values
M x N
Use data selection to select v samples
Obtain pruned IFT data $S_F \subset D_F$
$|S_F|$
Fine-tune model M'

NN-CIFT → Step 1: collect training data

Use existing influence function to calculate a few influence scores

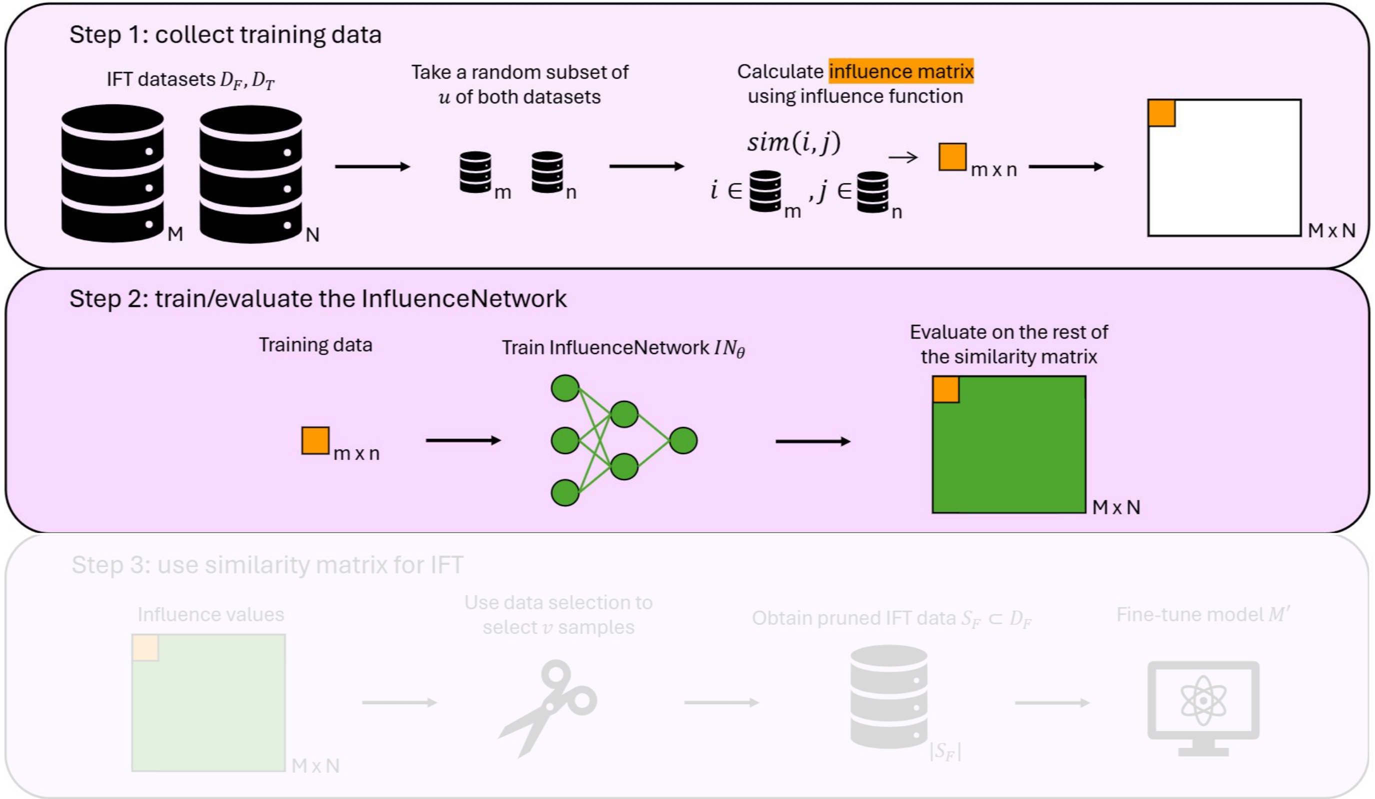


sim(i, j) = Influence of j on i

Our evaluation uses:

1. DELIFT [Agarwal et al. 2025]
2. + DELIFT (SE)
3. LESS [Xia et al. 2024]
4. SelectIT [Liu et al. 2024]

NN-CIFT


Step 1: collect training data
IFT datasets D_F, D_T
M
N
Take a random subset of
u of both datasets
m
n
Calculate influence matrix
using influence function
$sim(i,j)$
$i \in$ m $, j \in$ n
m x n
M x N
Step 2: train/evaluate the InfluenceNetwork
Training data
m x n
Train InfluenceNetwork IN_θ
Evaluate on the rest of
the similarity matrix
M x N
Step 3: use similarity matrix for IFT
Influence values
M x N
Use data selection to
select v samples
Obtain pruned IFT data $S_F \subset D_F$
$|S_F|$
Fine-tune model M'

NN-CIFT → Step 2: train the InfluenceNetwork

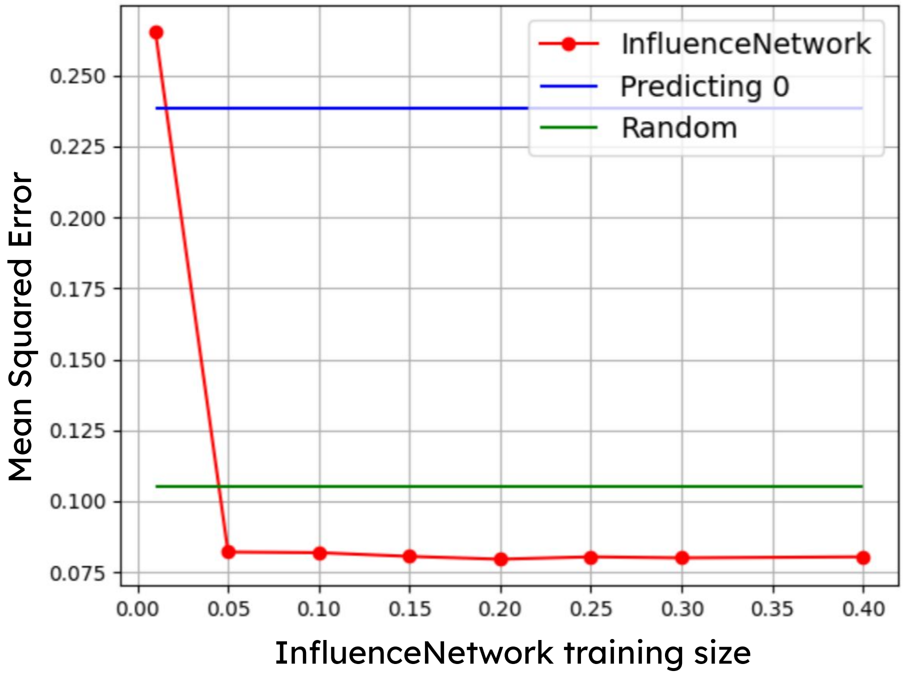
InfluenceNetwork definition:

```
class InfluenceNetwork(nn.Module):
    def __init__(self, dim, hidden_size=100):
        super(InfluenceNetwork, self).__init__()
        self.fc1 = nn.Linear(dim, hidden_size)      ← [2049 x 100]
        self.activate = nn.ReLU()
        self.fc2 = nn.Linear(hidden_size, 1)        ← [100 x 1]
```

Use mini-batch gradient descent to train

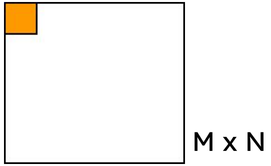
NN-CIFT → Step 2: train the InfluenceNetwork → Does it work well?

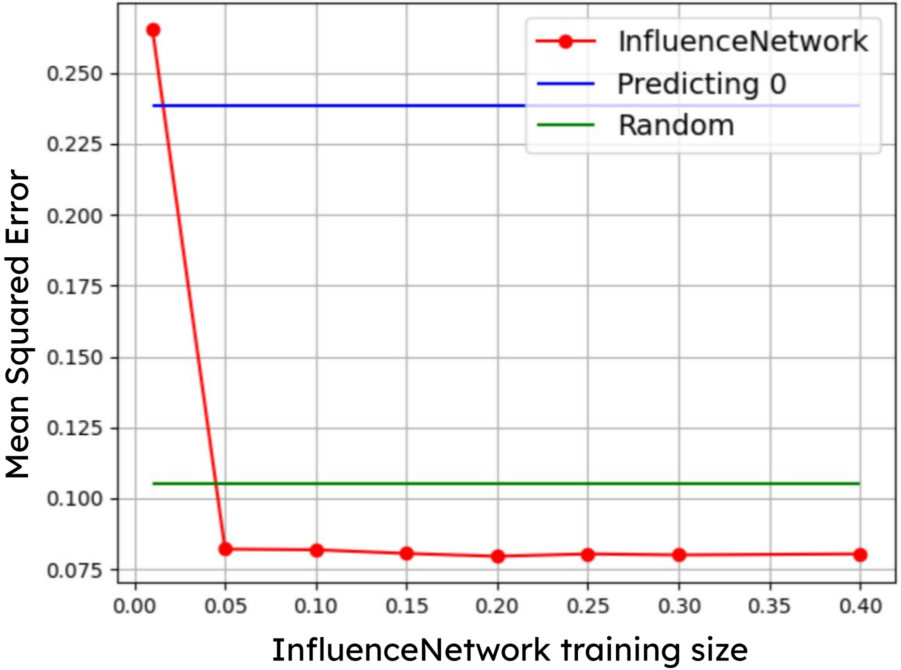
MSE on validation: 0.072!



NN-CIFT → Step 2: train the InfluenceNetwork → Does it work well?

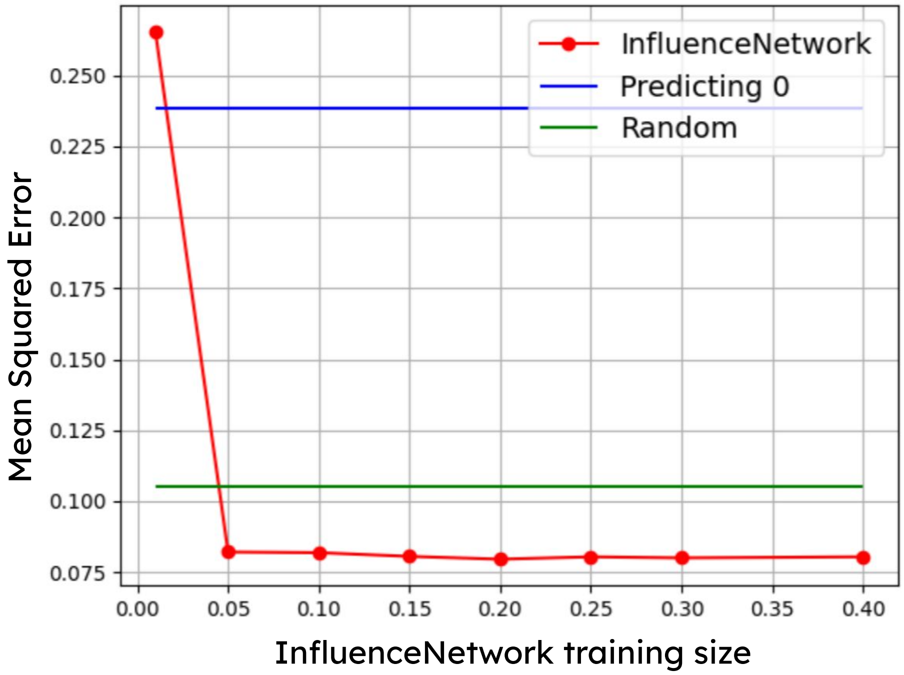
MSE on validation: 0.072!





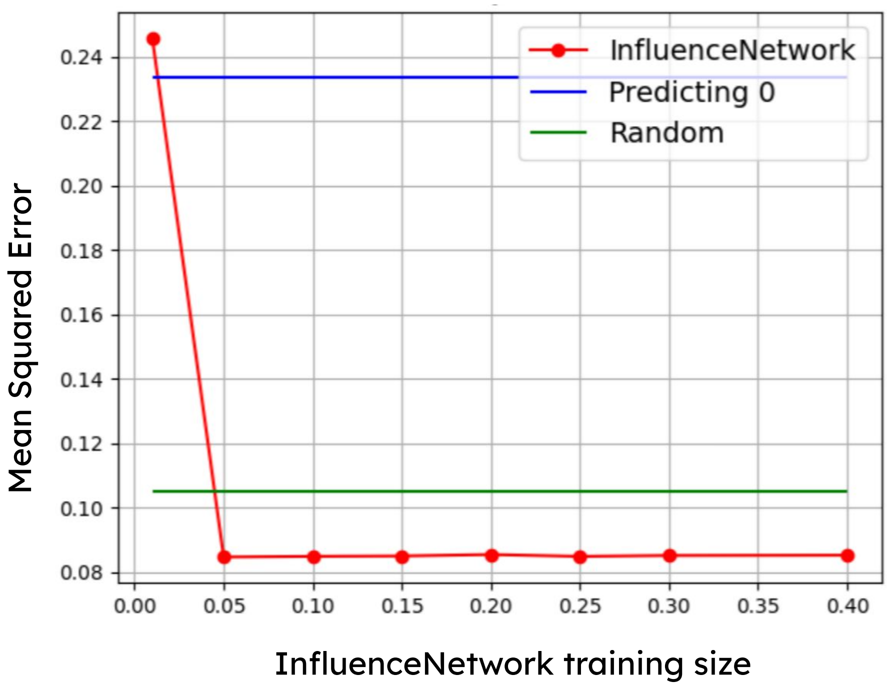
NN-CIFT → Step 2: train the InfluenceNetwork → Does it work well?

MSE on validation: 0.072!



NN-CIFT → Step 2: train the InfluenceNetwork → Does it work well?

MSE on test set: 0.063



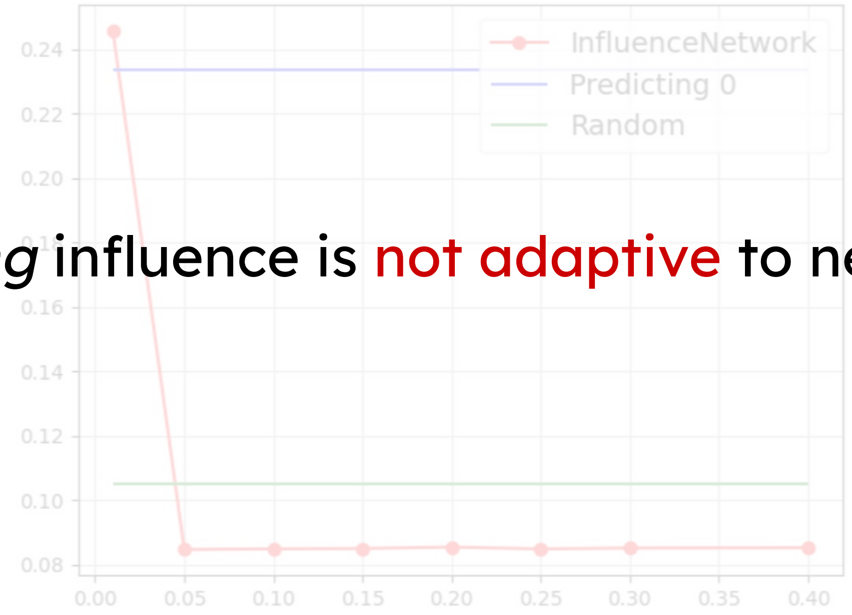
NN-CIFT → Step 2: train the InfluenceNetwork → Does it work well?

MSE on test set: 0.063

Why is this so cool?

NN-CIFT → Step 2: train the InfluenceNetwork → Does it work well?

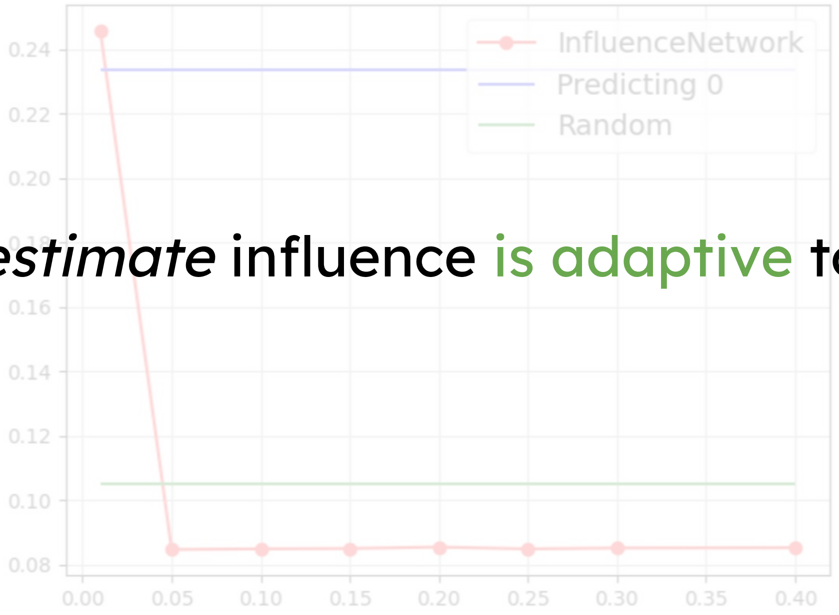
MSE on test set: 0.063



Estimating influence is not adaptive to new data

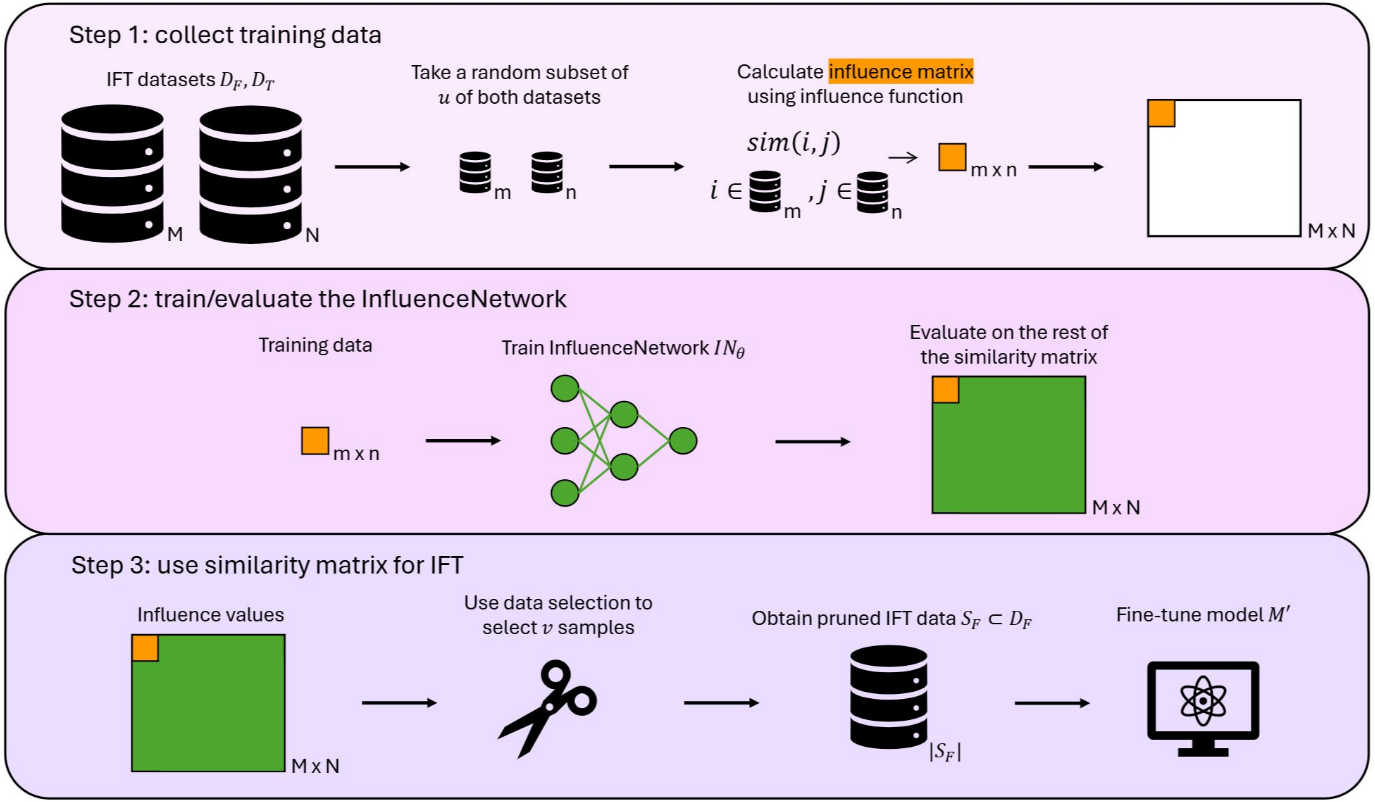
NN-CIFT → Step 2: train the InfluenceNetwork → Does it work well?

MSE on test set: 0.063



Learning to estimate influence is adaptive to new data

NN-CIFT


Step 1: collect training data
IFT datasets D_F, D_T
M
N
Take a random subset of u of both datasets
m
n
Calculate influence matrix using influence function
$sim(i,j)$
$i \in$ m $, j \in$ n
m x n
M x N
Step 2: train/evaluate the InfluenceNetwork
Training data
m x n
Train InfluenceNetwork IN_θ
Evaluate on the rest of the similarity matrix
M x N
Step 3: use similarity matrix for IFT
Influence values
M x N
Use data selection to select v samples
Obtain pruned IFT data $S_F \subset D_F$
$|S_F|$
Fine-tune model M'

NN-CIFT → Step 3: use sim. matrix for IFT → does pruning work?

CONVAI

Results on Phi-3

Dataset	MixInstruct						Alpaca					
Method	ICL			QLoRA			ICL			QLoRA		
Metric	ROUGE	BGE	LAJ	ROUGE	BGE	LAJ	ROUGE	BGE	LAJ	ROUGE	BGE	LAJ
Initial	37.87	78.92	2.98	36.36	82.55	3.02	25.79	67.82	2.56	27.29	71.57	2.62
Random	39.00	80.66	3.12	44.45	85.46	3.12	34.93	73.50	3.07	35.57	75.16	2.96
SelectIT	43.08	84.50	3.18	45.14	85.88	3.21	33.56	77.10	3.12	34.04	78.10	3.21
NN-CIFT + SelectIT	43.71	81.95	3.16	46.09	86.13	3.19	34.85	77.79	3.13	34.07	78.11	3.16
LESS	42.08	83.24	3.26	45.16	84.95	3.28	35.78	76.84	3.16	35.28	76.49	3.15
NN-CIFT + LESS	42.84	83.74	3.26	45.18	84.63	3.26	36.12	77.11	3.16	36.49	75.75	3.16
DELIFT (SE)	47.43	84.40	3.28	48.22	86.50	3.28	37.53	80.76	3.25	42.66	84.26	3.18
NN-CIFT + DELIFT (SE)	47.30	82.99	3.23	46.49	84.68	3.29	37.02	80.72	3.26	42.52	84.58	3.29
DELIFT	48.46	85.77	3.35	52.79	88.04	3.37	38.36	81.13	3.36	43.43	85.05	3.56
NN-CIFT + DELIFT	48.57	83.90	3.41	53.30	81.34	3.54	38.99	80.29	3.49	44.64	85.23	3.57
Full Data	58.65	88.72	3.45	65.51	92.24	3.51	35.27	77.85	3.31	39.29	78.85	3.29

NN-CIFT → Step 3: use sim. matrix for IFT → does pruning work?

CONVAI

Results on Phi-3

Dataset	MixInstruct						Alpaca					
Method	ICL			QLoRA			ICL			QLoRA		
Metric	ROUGE	BGE	LAJ	ROUGE	BGE	LAJ	ROUGE	BGE	LAJ	ROUGE	BGE	LAJ
Initial	37.87	78.92	2.98	36.36	82.55	3.02	25.79	67.82	2.56	27.29	71.57	2.62
Random	39.00	80.66	3.12	44.45	85.46	3.12	34.93	73.50	3.07	35.57	75.16	2.96
SelectIT	43.08	84.50	3.18	45.14	85.88	3.21	33.56	77.10	3.12	34.04	78.10	3.21
NN-CIFT + SelectIT	43.71	81.95	3.16	46.09	86.13	3.19	34.85	77.79	3.13	34.07	78.11	3.16
LESS	42.08	83.24	3.26	45.16	84.95	3.28	35.78	76.84	3.16	35.28	76.49	3.15
NN-CIFT + LESS	42.84	83.74	3.26	45.18	84.63	3.26	36.12	77.11	3.16	36.49	75.75	3.16
DELIFT (SE)	47.43	84.40	3.28	48.22	86.50	3.28	37.53	80.76	3.25	42.66	84.26	3.18
NN-CIFT + DELIFT (SE)	47.30	82.99	3.23	46.49	84.68	3.29	37.02	80.72	3.26	42.52	84.58	3.29
DELIFT	48.46	85.77	3.35	52.79	88.04	3.37	38.36	81.13	3.36	43.43	85.05	3.56
NN-CIFT + DELIFT	48.57	83.90	3.41	53.30	81.34	3.54	38.99	80.29	3.49	44.64	85.23	3.57
Full Data	58.65	88.72	3.45	65.51	92.24	3.51	35.27	77.85	3.31	39.29	78.85	3.29

Difference between NN-CIFT and original influence function is 1.40%!

NN-CIFT → Step 3: use sim. matrix for IFT → does pruning work?

CONVAI

Results on Llama-8B

Dataset	MixInstruct						Alpaca					
Method	ICL			QLoRA			ICL			QLoRA		
Metric	ROUGE	BGE	LAJ	ROUGE	BGE	LAJ	ROUGE	BGE	LAJ	ROUGE	BGE	LAJ
Initial	28.53	74.05	2.94	34.42	78.54	3.00	24.85	72.45	2.26	34.29	80.82	3.03
Random	40.07	84.04	3.26	41.68	84.26	3.22	36.95	80.47	3.12	38.64	80.46	3.07
SelectIT	46.51	86.18	3.25	50.31	87.38	3.25	41.42	83.25	3.27	44.51	84.18	3.34
NN-CIFT + SelectIT	46.48	85.86	2.28	50.87	87.43	3.26	42.07	83.67	3.27	44.99	85.13	3.37
LESS	48.21	86.19	3.34	51.24	86.07	3.37	43.34	84.19	3.38	44.73	84.04	3.32
NN-CIFT + LESS	48.20	86.31	3.36	51.56	86.39	3.41	44.42	84.69	3.32	46.40	85.44	3.36
DELIFT (SE)	48.36	85.91	3.38	51.43	86.20	3.34	44.30	85.52	3.41	45.35	86.34	3.48
NN-CIFT + DELIFT (SE)	48.59	85.01	3.39	50.53	86.10	3.33	45.49	86.27	3.44	45.75	86.45	3.47
DELIFT	51.66	88.02	3.43	55.58	91.81	3.50	46.49	87.60	3.50	49.16	87.74	3.54
NN-CIFT + DELIFT	52.03	88.38	3.41	55.85	91.96	3.51	46.26	87.41	3.55	49.15	87.74	3.50
Full Data	54.43	92.55	3.40	59.47	94.12	3.58	48.53	91.21	3.63	48.29	90.82	3.66

Here, its 1.39%!

NN-CIFT → Step 3: use sim. matrix for IFT → what about the cost?

Costs (seconds) are cut down by 77% to 99%

Model	Phi-3		Llama-8B	
Dataset	MixInstruct	Alpaca	MixInstruct	Alpaca
Initial	-	-	-	-
Random	12.4	12.3	12.9	12.3
SelectIT	7,047	6,594	6,671	6,470
NN-CIFT + SelectIT	65	63	64	63
LESS	12,338	11,217	10,843	14,819
NN-CIFT + LESS	78	75	74	84
DELIFT (SE)	216	218	218	219
NN-CIFT + DELIFT (SE)	48	48	48	48
DELIFT	67,379	68,117	68,076	65,711
NN-CIFT + DELIFT	215	217	217	211
Full Data	-	-	-	-

What does this all mean?

NN-CIFT…

- uses neural networks instead of language models for data valuation
- learns to estimate influence with very low error
- scales with new data, which was not previously possible
- reduces up to 99% costs without affecting performance

Cheap Neural Networks for Learnable and Scalable Influence Estimation of Instruction Fine-Tuning

Ishika Agarwal, Dilek Hakkani-Tür

CONVAI

arXiv

GitHub

NN-CIFT...

Uses neural networks for data valuation – it is adaptive to new data and reduces costs by 99%

- Uses only 5% of data to train neural network
- Network size reduces to 0.0027% of LLMs
- Accurate data selection (MSE: 0.067)

How is influence estimation done?

- Model Dependent
  - Inference-based
    - DELIFT: quantifies how close the answer is to the ground truth
    - SelectIT: uses confidence and consistency to estimate importance
  - Gradient-based
    - LESS: measures how similar the gradients are for two data points
- Model Independent
  - Clustering-based: data points that match a target task are better to learn

Pairwise v Pointwise

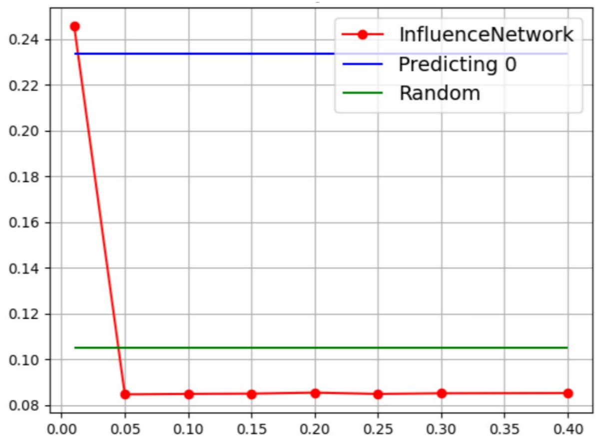
- Pairwise: uses the mutual/conditional information between two datasets
    - IFT: does Common Corpus already have this knowledge from MixInstruct?
    - Task-specific FT: what data from MixInstruct looks like GSM8k?
    - Continual learning: what data from DatasetName-v2 is different from DatasetName-v1?
- Pointwise: computes a ranking of “importance”

Evaluation Details: InfluenceNetwork

- Dataset: MixInstruct
- Model: Phi-3
- Influence Function: DELIFT
- MSE:

$$\frac{1}{|\mathcal{D}_{\mathcal{F}} \times \mathcal{D}_{\mathcal{T}}|} \sum_{(i,j) \in \mathcal{D}_{\mathcal{F}} \times \mathcal{D}_{\mathcal{T}}} (IF_\theta(i,j) - \text{sim}(i,j))^2$$

  - D_F → MixInstruct training
  - D_T → MixInstruct validation
  - IF_theta → InfluenceNetwork
  - sim(i,j) → DELIFT
- Baselines:
  - Predicting only 0 influence
  - Predicting random influence



Evaluation Details: subset selection

- Datasets: MixInstruct, Alpaca
- Model: Phi-3, Llama-8B
- Metrics:
  - ROUGE
  - BGE: cosine distance between BAAI General Embeddings
  - LAJ: Llama-as-a-Judge, Prometheus
- ICL/QLoRA:
  - ICL: use chosen subset as pool for ICL
  - QLoRA: fine-tune on chosen subset
- Baselines:
  - Initial: no subset selection
  - Random: selecting a random subset
  - Original influence function
  - Using DistilGPT2 as the LM in IFs
  - Full Data: using 100% of the data

Dataset	MixInstruct						Alpaca					
Method	ICL			QLoRA			ICL			QLoRA		
Metric	ROUGE	BGE	LAJ	ROUGE	BGE	LAJ	ROUGE	BGE	LAJ	ROUGE	BGE	LAJ
Initial	28.53	74.05	2.94	34.42	78.54	3.00	24.85	72.45	2.26	34.29	80.82	3.03
Random	40.07	84.04	3.26	41.68	84.26	3.22	36.95	80.47	3.12	38.64	80.46	3.07
SelectIT	46.51	86.18	3.25	50.31	87.38	3.25	41.42	83.25	3.27	44.51	84.18	3.34
DistilGPT2 + SelectIT	41.26	80.33	3.20	44.86	84.72	3.23	39.18	80.99	2.99	41.72	81.50	3.14
NN-CIFT + SelectIt	46.48	85.86	2.28	50.87	87.43	3.26	42.07	83.67	3.27	44.99	85.13	3.37
LESS	48.21	86.19	3.34	51.24	86.07	3.37	43.34	84.19	3.38	44.73	84.04	3.32
DistilGPT2 + LESS	42.18	78.34	3.23	48.64	79.09	3.27	42.02	80.89	3.29	42.51	82.35	3.29
NN-CIFT + LESS	48.20	86.31	3.36	51.56	86.39	3.41	44.42	84.69	3.32	46.40	85.44	3.36
DELIFT (SE)	48.36	85.91	3.38	51.43	86.20	3.34	44.30	85.52	3.41	45.35	86.34	3.48
DistilGPT2 + DELIFT (SE)	47.21	84.24	3.28	49.37	84.24	3.29	43.51	85.45	3.41	44.89	79.81	3.36
NN-CIFT + DELIFT (SE)	48.59	85.01	3.39	50.53	86.10	3.33	45.49	86.27	3.44	45.75	86.45	3.47
DELIFT	51.66	88.02	3.43	55.58	91.81	3.50	46.49	87.60	3.50	49.16	87.74	3.54
DistilGPT2 + DELIFT	47.09	84.74	3.26	48.21	84.24	3.28	45.08	81.45	3.41	41.07	83.22	3.44
NN-CIFT + DELIFT	52.03	88.38	3.41	55.85	91.96	3.51	46.26	87.41	3.55	49.15	87.74	3.50
Full Data	54.43	92.55	3.40	59.47	94.12	3.58	48.53	91.21	3.63	48.29	90.82	3.66