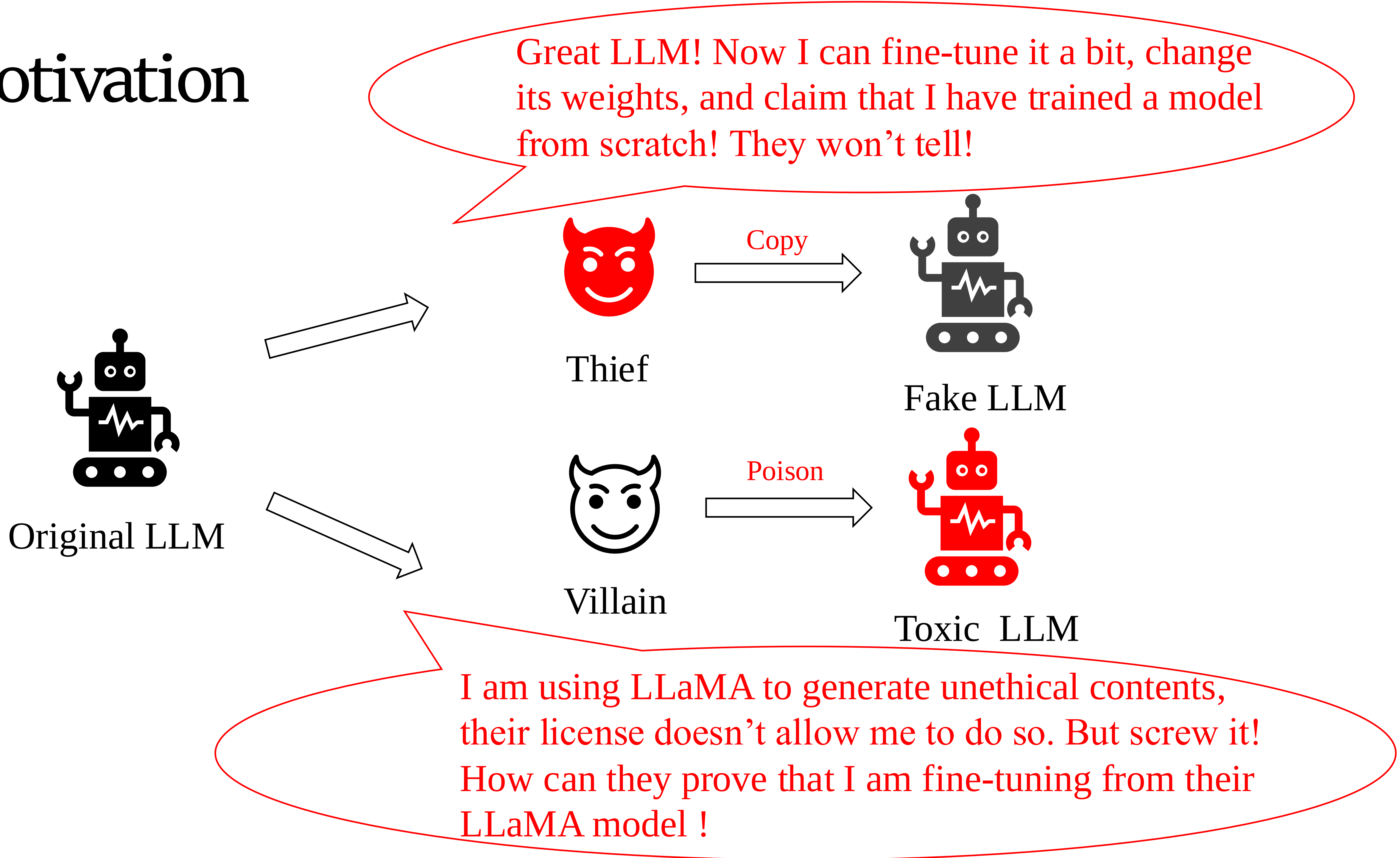
上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

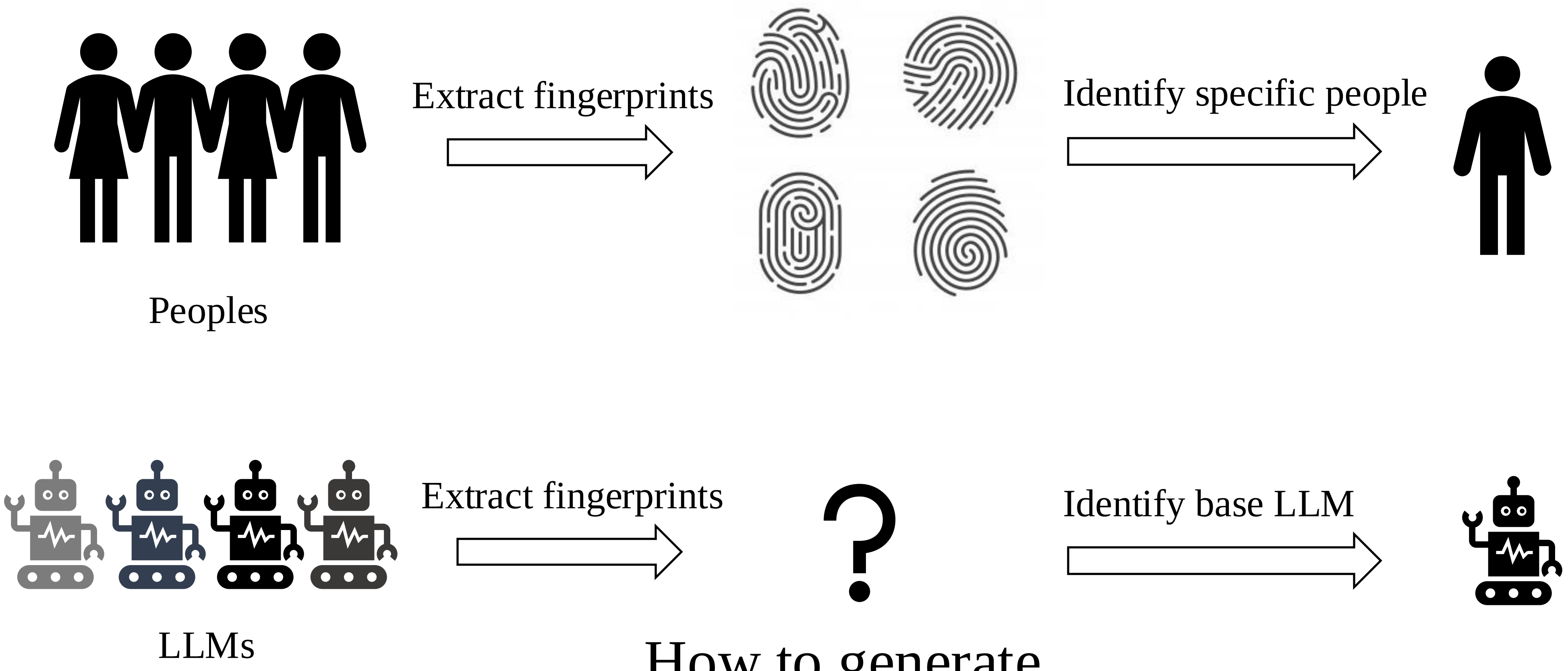
HuRef: HUman-REadable Fingerprint for Large Language Models

Boyi Zeng, Lizheng Wang, Yuncong Hu, Yi Xu,
Chenghu Zhou, Xinbing Wang, Yu Yu, Zhouhan Lin*

Motivation

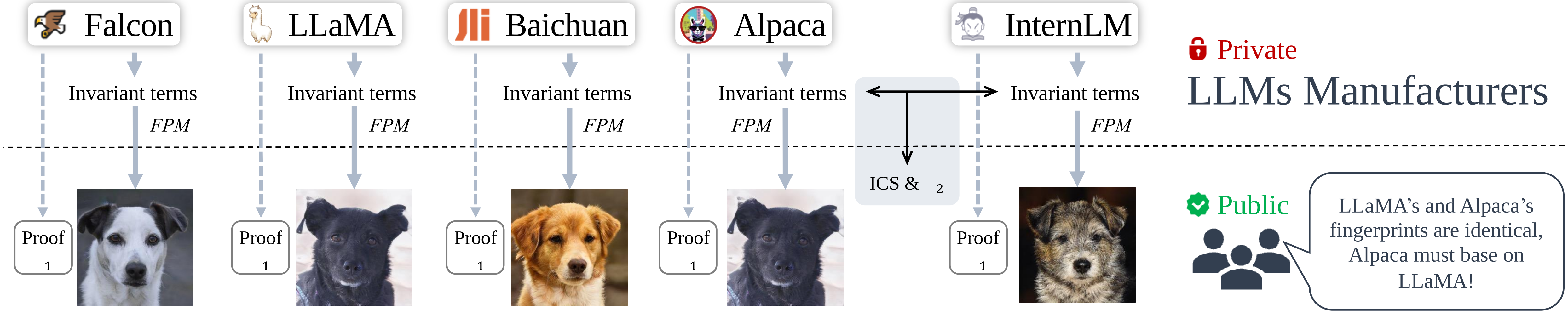

Great LLM! Now I can fine-tune it a bit, change its weights, and claim that I have trained a model from scratch! They won't tell!
Copy
Thief
Fake LLM
Original LLM
Poison
Villain
Toxic LLM
I am using LLaMA to generate unethical contents, their license doesn't allow me to do so. But screw it! How can they prove that I am fine-tuning from their LLaMA model !

Motivation



How to generate
fingerprints for LLMs?

Our solution to LLM fingerprints



1. LLM manufacturers extract invariant terms of their model.
2. LLM manufacturers use a fingerprinting model to generate fingerprint images and publish them.
3. At the same time, they generate and publish zero-knowledge proofs for the extraction of invariant terms and the fingerprinting process.
4. The public identifies LLMs' base model according to their fingerprint images, and can verify through the zero-knowledge proof whether the fingerprints were honestly generated.

Protecting LLMs based on fingerprints without revealing model parameters.

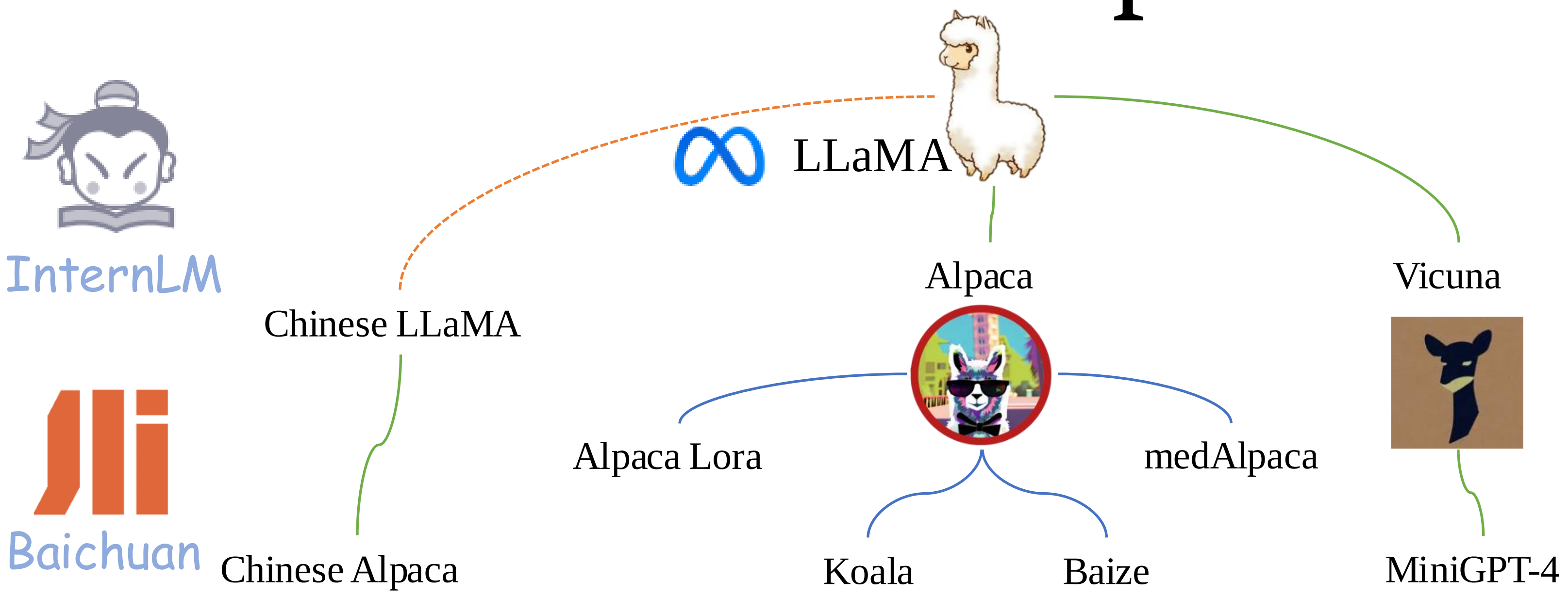
Our observation on LLM parameters

LLM parameters vector: $Concat(\bigcup_i flatten\,(W_i)),\quad W_i \in LLM\ weights.$

LLM vectors:

LLaMA	Alpaca	Vicuna	OpenLLaMA
1	0.99	1.01	0.01
0	0.01	-0.01	1.0
1	0.98	0.99	0
0	-0.02	0.02	0.99
1	1.03	1	-0.01
0	0.01	-0.01	-1

Our observation on LLM parameters



Model	Alpaca-Lora	Alpaca	Chinese-LLaMA	Vicuna	Baize	MedAlpaca	Koala	WizardLM	MiniGPT-4	Chinese-Alpaca	Baichuan	OpenLLaMA	InternLM	LLaMA-2
PCS	99.87	99.91	99.68	99.80	99.73	99.90	99.82	99.89	99.70	99.52	0.83	1.16	0.28	1.51

PCS is short for “parmeter cosine similarity”, which is the cosine similarities of model parameters between various LLMs w.r.t. the LLaMA-7B base model.

LLaMA’s offspring models maintain high PCS w.r.t the LLaMA-7B base model, while independently pretrained LLMs showing almost zero cosine similarity with the LLaMA-7B model.

Our observation on LLM parameters

1. The vector direction of LLM parameters remains stable through subsequent training steps, including continued pretraining, supervised fine-tuning (SFT), and RLHF. (high cosine similarity)

2. Independently pretrained LLMs showing clearly different parameters' vector direction. (almost zero cosine similarity)

Model	Alpaca-Lora	Alpaca	Chinese-LLaMA	Vicuna	Baize	MedAlpaca	Koala	WizardLM	MiniGPT-4	Chinese-Alpaca	Baichuan	OpenLLaMA	InternLM	LLaMA-2
PCS	99.87	99.91	99.68	99.80	99.73	99.90	99.82	99.89	99.70	99.52	0.83	1.16	0.28	1.51

We can calculate cosine similarities of LLM parameters' vectors to identify their base model!

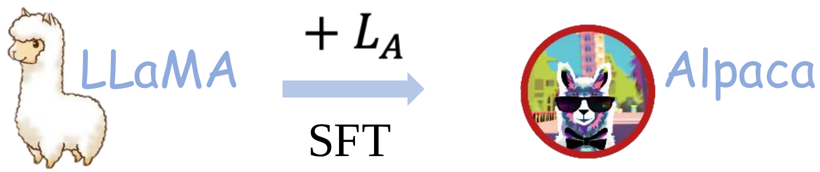
PCS is short for "parmeter cosine similarity", which is the cosine similarities of model parameters between various LLMs w.r.t. the LLaMA-7B base model.

Our observation on LLM parameters

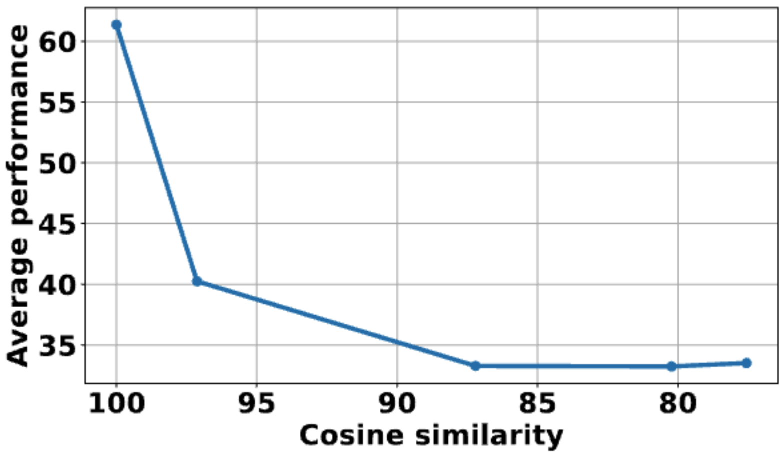
This is great! But to make this principle robust to intentional attacks, we need to know how hard it is to circumvent this principle?

i.e., would it be easy for someone to intentionally alter the parameter vector direction, while still maintaining the model's ability?

Our observation on LLM parameters



$$L = L_{orin} + L_A \qquad L_A = \frac{|\langle V_A, V_{base} \rangle|}{|V_A||V_{base}|}$$



The model's performance quickly deteriorates as the cosine similarity decreases.

Model	BoolQ	HellaSwag	PIQA	WinoGrande	ARC-e	ARC-c	RACE	MMLU	Avg.
LLaMA	75.11	76.19	79.16	70.00	72.90	44.80	40.00	32.75	61.36
Alpaca	77.49	75.64	77.86	67.80	70.66	46.58	43.16	41.13	62.54
$+L_A$(epoch1)	45.44	31.16	67.63	48.70	49.03	34.13	22.78	23.13	40.25
$+L_A$(epoch2)	42.23	26.09	49.78	47.43	26.43	28.92	22.97	23.22	33.38
$+L_A$(epoch3)	39.05	26.40	49.95	48.30	26.52	28.75	22.97	23.98	33.24
$+L_A$(epoch4)	41.62	26.15	50.11	49.33	26.56	28.50	22.78	23.12	33.52
$+L_A$(epoch5)	38.56	26.13	50.11	50.20	26.22	29.10	22.39	27.02	33.72

Table 3: Zero-shot performance on multiple standard benchmarks.

It's fairly hard to deviate the model parameter's vector direction without damaging the base model's abilities!

From parameter vector direction to invariant terms

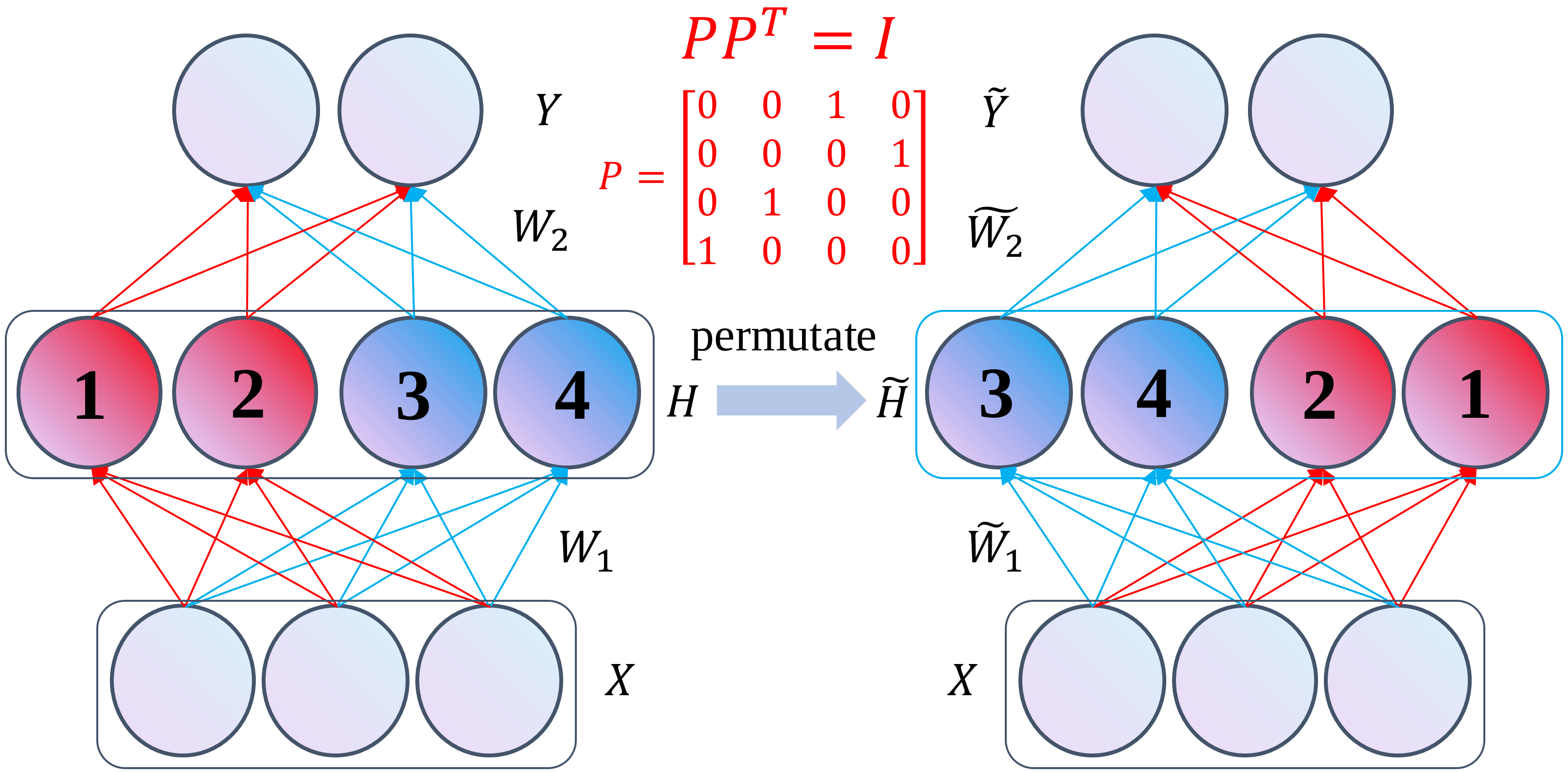
Parameter vector direction is a good indicator for identifying the base model for LLM! It is both reliable and robust.

……But wait a second, directly using the parameter vector direction has problems.

1. It requires to reveal the model parameter directly, which is not always acceptable in this LLM era.
2. Attackers can perform **weight rearrangemet attacks** to the model, by permutating hidden units.

An example of weight rearrangement attack: Permutation Attack

Taking a simple FFN of transformer as an example:



$$Y = \sigma(XW_1 + b_1)W_2 + b_2$$

$$\tilde{Y} = \sigma(X\widetilde{W}_1 + \widetilde{b_1})\widetilde{W}_2 + \widetilde{b_2}$$

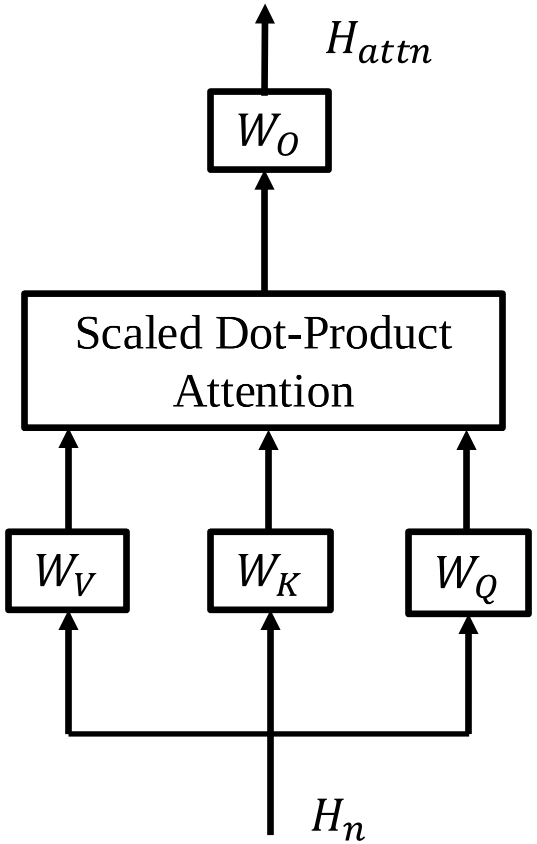
$$= \sigma(XW_1P + b_1P)P^TW_2 + P^Tb_2$$

$$= Y$$

We can easily change the parameters' (W_1, W_2) direction through permutating hidden units in H without affecting output (Y).

Linear mapping Attack

For attention layer of transformer (single head) :



$$\boxed{H_{Attn} = softmax\left(\frac{H_n W_Q W_K^T H_n^T}{\sqrt{d}}\right) H_n W_V W_O}$$

For any invertible matrix C_1, C_2:

$$\widetilde{W_Q} = W_Q C_1 \quad \widetilde{W_K} = C_1^{-1} W_K^T \quad \widetilde{W_V} = W_V C_2 \quad \widetilde{W_O} = C_2^{-1} W_O$$

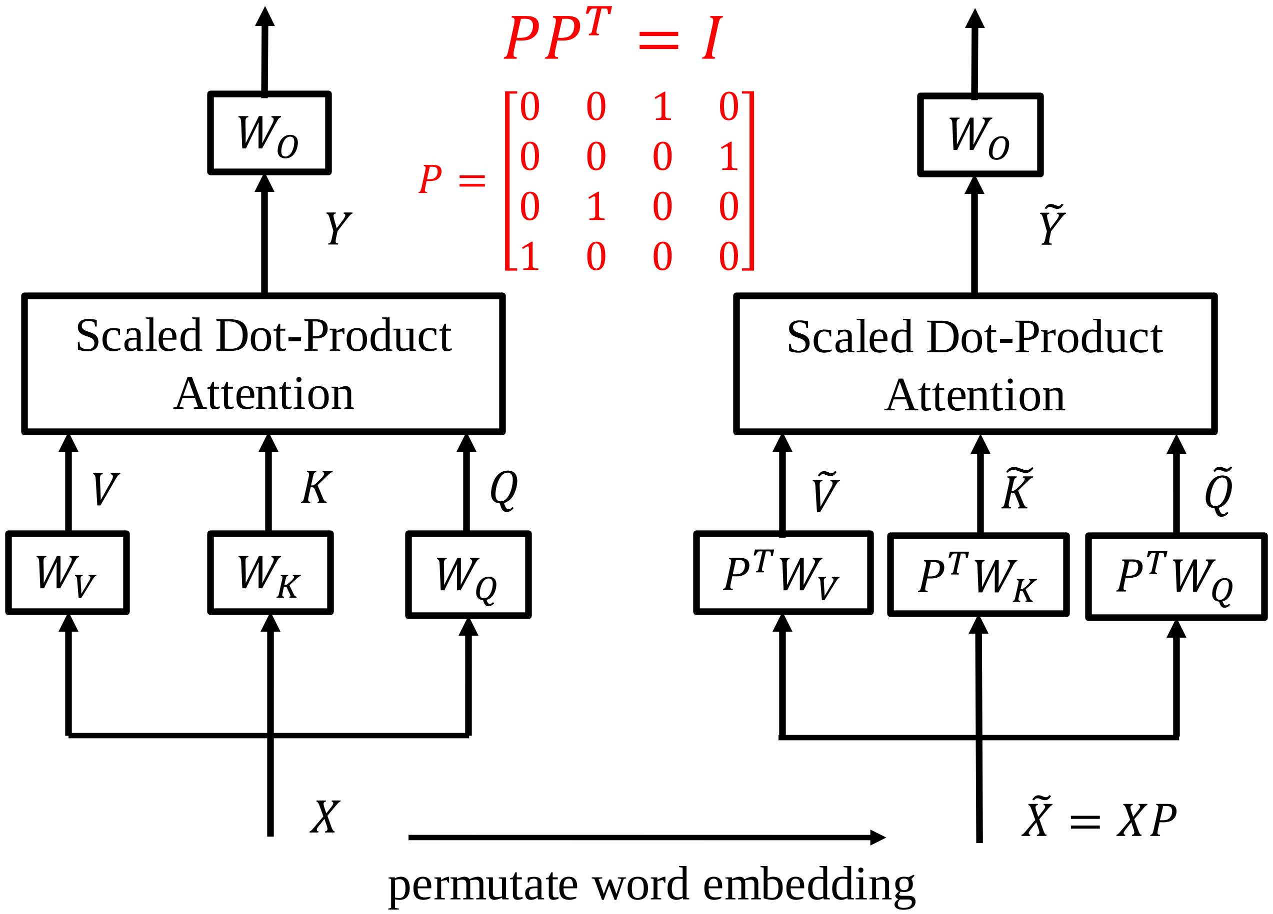
$$\widetilde{H_{Attn}} = softmax\left(\frac{H_n (W_Q C_1)(C_1^{-1} W_K^T) H_n^T}{\sqrt{d}}\right) H_n (W_V C_2)(C_2^{-1} W_O)$$
$$= H_{Attn}$$

$$\langle W_Q, \widetilde{W_Q} \rangle \neq 1 \quad \langle W_K, \widetilde{W_K} \rangle \neq 1 \quad \langle W_V, \widetilde{W_V} \rangle \neq 1 \quad \langle W_O, \widetilde{W_O} \rangle \neq 1$$

We can change the parameters' (W_Q, W_K, W_V, W_O) direction through linear mapping without affecting output (H_{Attn}).

Permutation Attack on word embeddings

For attention layer of transformer (single head) :



$$PP^T = I$$

$$P = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

$$\tilde{V} = XPP^T W_V = \mathrm{V}$$

$$\tilde{K} = XPP^T W_K = \mathrm{K}$$

$$\tilde{Q} = XPP^T W_Q = \mathrm{Q}$$

$$\downarrow$$

$$\tilde{Y} = Y$$

$$\langle X, \tilde{X} \rangle \neq 1 \quad \langle W_Q, \widetilde{W_Q} \rangle \neq 1$$

$$\langle W_K, \widetilde{W_K} \rangle \neq 1 \quad \langle W_V, \widetilde{W_V} \rangle \neq 1$$

We can change the parameters' (X, W_Q, W_K, W_V) direction by jointly permutating dimensions in word embeddings X and W_Q, W_K, W_V.

Forms of Weight Rearrangement Attacks

Principle: Change vector direction without changing architecture or affecting output.

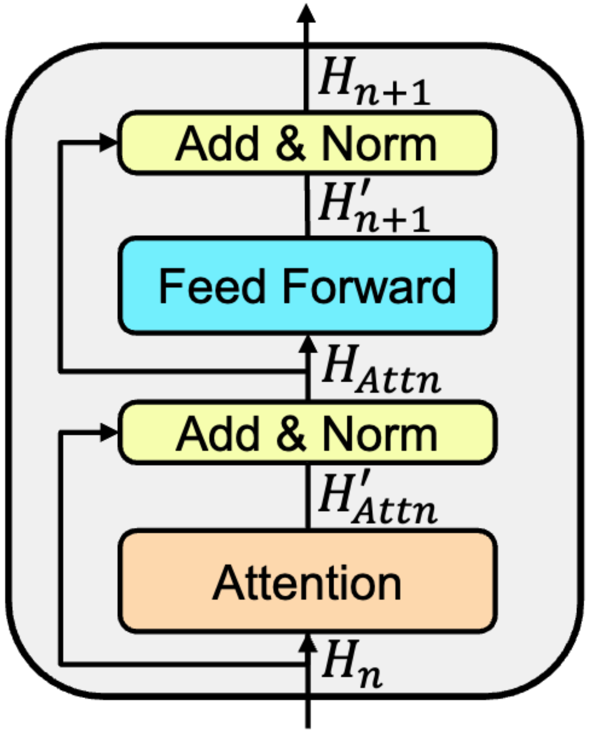


Figure 2: Transformer layer

$$\boldsymbol{H}'_{Attn} = \mathrm{softmax}\left(\frac{\boldsymbol{H}_n \boldsymbol{W}_Q (\boldsymbol{H}_n \boldsymbol{W}_K)^T}{\sqrt{d}}\right) \boldsymbol{H}_n \boldsymbol{W}_V \boldsymbol{W}_O$$

$$\boldsymbol{H}'_{n+1} = \sigma\left(\boldsymbol{H}_{Attn} \boldsymbol{W}_1 + \mathrm{b}_1\right) \boldsymbol{W}_2 + \mathrm{b}_2$$

1. Linear mapping attack on $\boldsymbol{W}_Q, \boldsymbol{W}_K$ and $\boldsymbol{W}_V, \boldsymbol{W}_O$.

$$\tilde{\boldsymbol{W}}_Q = \boldsymbol{W}_Q \boldsymbol{C}_1, \quad \tilde{\boldsymbol{W}}_K = \boldsymbol{W}_K \boldsymbol{C}_1^{-1}$$

2. Permutation attack on $\boldsymbol{W}_1, \mathrm{b}_1, \boldsymbol{W}_2$.

$$\tilde{\boldsymbol{W}}_1 = \boldsymbol{W}_1 \boldsymbol{P}_{FFN}, \quad \tilde{\boldsymbol{W}}_2 = \boldsymbol{P}_{FFN}^{-1} \boldsymbol{W}_2, \quad \tilde{\mathrm{b}_1} = \mathrm{b}_1 \boldsymbol{P}_{FFN}$$

3. Permutation attack on word embeddings.

$$\tilde{\boldsymbol{X}} = \boldsymbol{X}\boldsymbol{P}_E, \tilde{\boldsymbol{W}}_1 = \boldsymbol{P}_E^{-1} \boldsymbol{W}_1, \quad \tilde{\boldsymbol{W}}_2 = \boldsymbol{W}_2 \boldsymbol{P}_E, \quad \tilde{\mathrm{b}_2} = \mathrm{b}_2 \boldsymbol{P}_E$$

$$\tilde{\boldsymbol{W}}_Q = \boldsymbol{P}_E^{-1} \boldsymbol{W}_Q, \quad \tilde{\boldsymbol{W}}_K = \boldsymbol{P}_E^{-1} \boldsymbol{W}_K, \quad \tilde{\boldsymbol{W}}_V = \boldsymbol{P}_E^{-1} \boldsymbol{W}_V, \quad \tilde{\boldsymbol{W}}_O = \boldsymbol{W}_O \boldsymbol{P}_E$$

Invariant Terms

Put attacks together:

$$\tilde{\boldsymbol{W}}_Q = \boldsymbol{P}_E^{-1}\boldsymbol{W}_Q\boldsymbol{Q}_1, \quad \tilde{\boldsymbol{W}}_K = \boldsymbol{P}_E^{-1}\boldsymbol{W}_K\boldsymbol{Q}_1^{-T}, \quad \tilde{\boldsymbol{W}}_V = \boldsymbol{P}_E^{-1}\boldsymbol{W}_V\boldsymbol{Q}_2, \quad \tilde{\boldsymbol{W}}_O = \boldsymbol{Q}_2^{-1}\boldsymbol{W}_O\boldsymbol{P}_E$$

$$\tilde{\boldsymbol{W}}_1 = \boldsymbol{P}_E^{-1}\boldsymbol{W}_1\boldsymbol{P}_{FFN}, \quad \tilde{\mathrm{b}_1} = \mathrm{b}_1\boldsymbol{P}_{FFN}, \quad \tilde{\boldsymbol{W}}_2 = \boldsymbol{P}_{FFN}^{-1}\boldsymbol{W}_2\boldsymbol{P}_E, \quad \tilde{\mathrm{b}_2} = \mathrm{b}_2\boldsymbol{P}_E \quad \tilde{\boldsymbol{X}} = \boldsymbol{X}\boldsymbol{P}_E, \quad \tilde{\boldsymbol{E}} = \boldsymbol{P}_E^{-1}\boldsymbol{E}$$

Eliminating attack matrices through multiplication.

Construct 3 invariant terms:

$$\boldsymbol{M}_a = \hat{\boldsymbol{X}}\boldsymbol{W}_Q\boldsymbol{W}_K^T\hat{\boldsymbol{X}}^T, \quad \boldsymbol{M}_b = \hat{\boldsymbol{X}}\boldsymbol{W}_V\boldsymbol{W}_O\hat{\boldsymbol{X}}^T, \quad \boldsymbol{M}_f = \hat{\boldsymbol{X}}\boldsymbol{W}_1\boldsymbol{W}_2\hat{\boldsymbol{X}}^T$$

Procedures to get $\hat{X}$:

1. Select a sufficiently big corpus as a standard verifying corpus.
2. Tokenize the corpus with the LLM's own vocabulary, and sort all tokens in the vocabulary according to their frequency.
3. Delete all tokens in the vocabulary that don't show up in the corpus.
4. Among the remaining tokens, select the least frequent K tokens as the tokens to be included in $\hat{\mathbf{X}}$.

From invariant terms to human-readable fingerprint

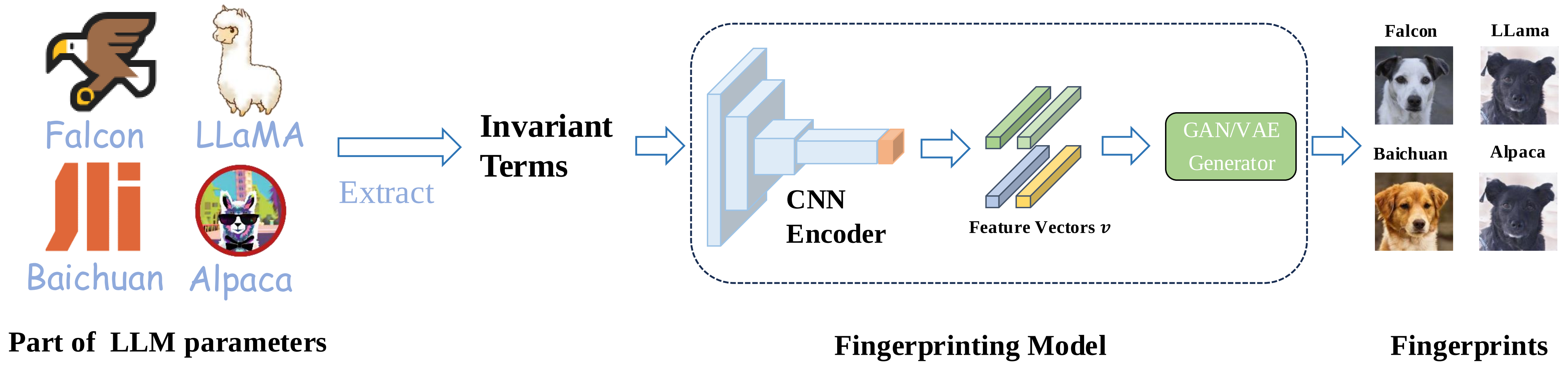
Can we use the invariant terms of LLM as its fingerprint?

No, publishing invariant terms may leak hidden information, including statistical features and parameter distributions. For example, the hidden size can be inferred through the rank of invariant terms.

We need to mitigate the risk of leakage while providing better visualization by making the invariant terms human-readable.

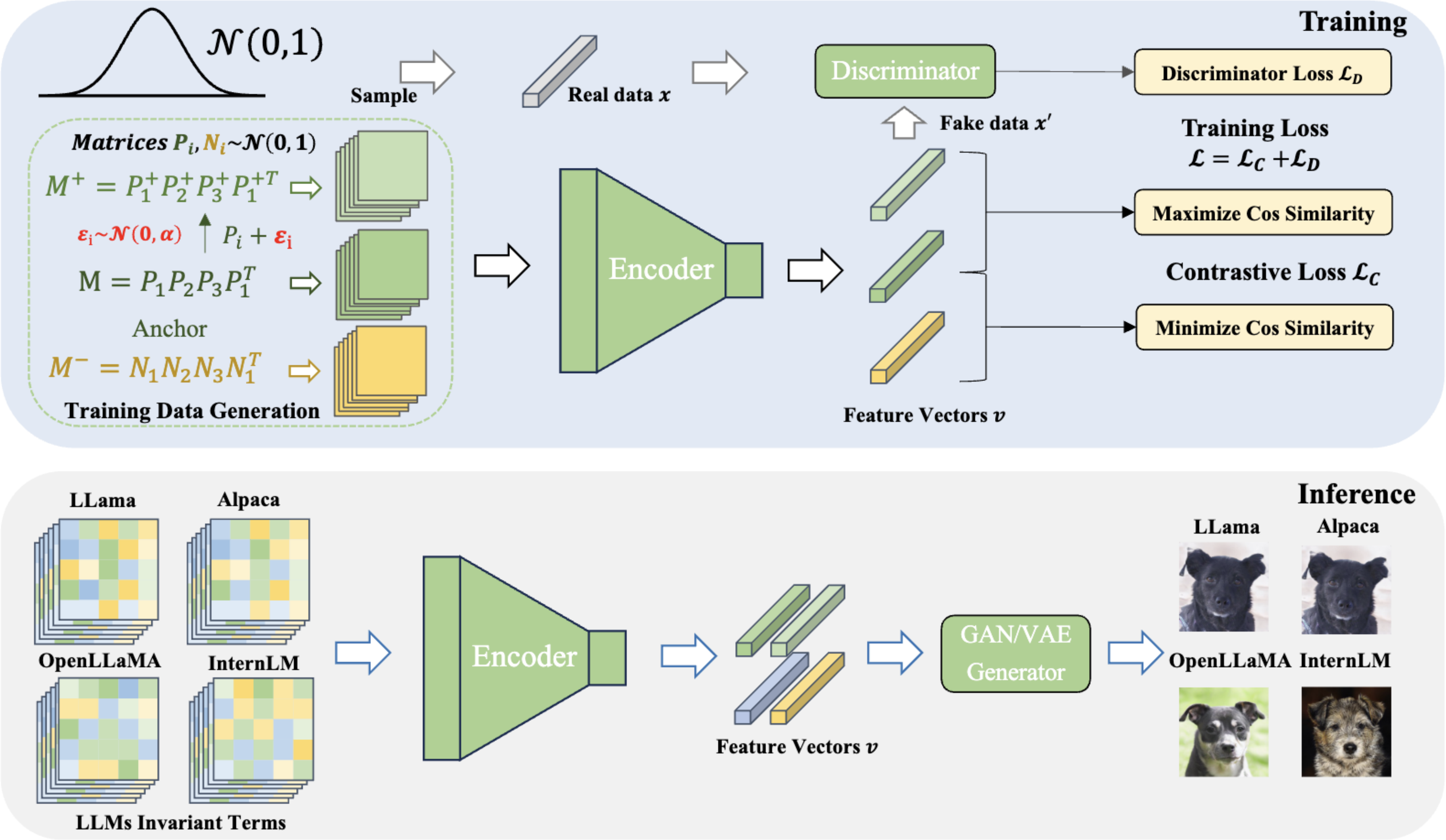
Generate human-readable fingerprint for LLMs

- Encode invariant terms to feature vectors through convnets.
- Mapping feature vectors to dog images using VAE or GAN generators.



Similar dogs share same base model, and vice versa.

Traning & inference framework for the fingerprinting model

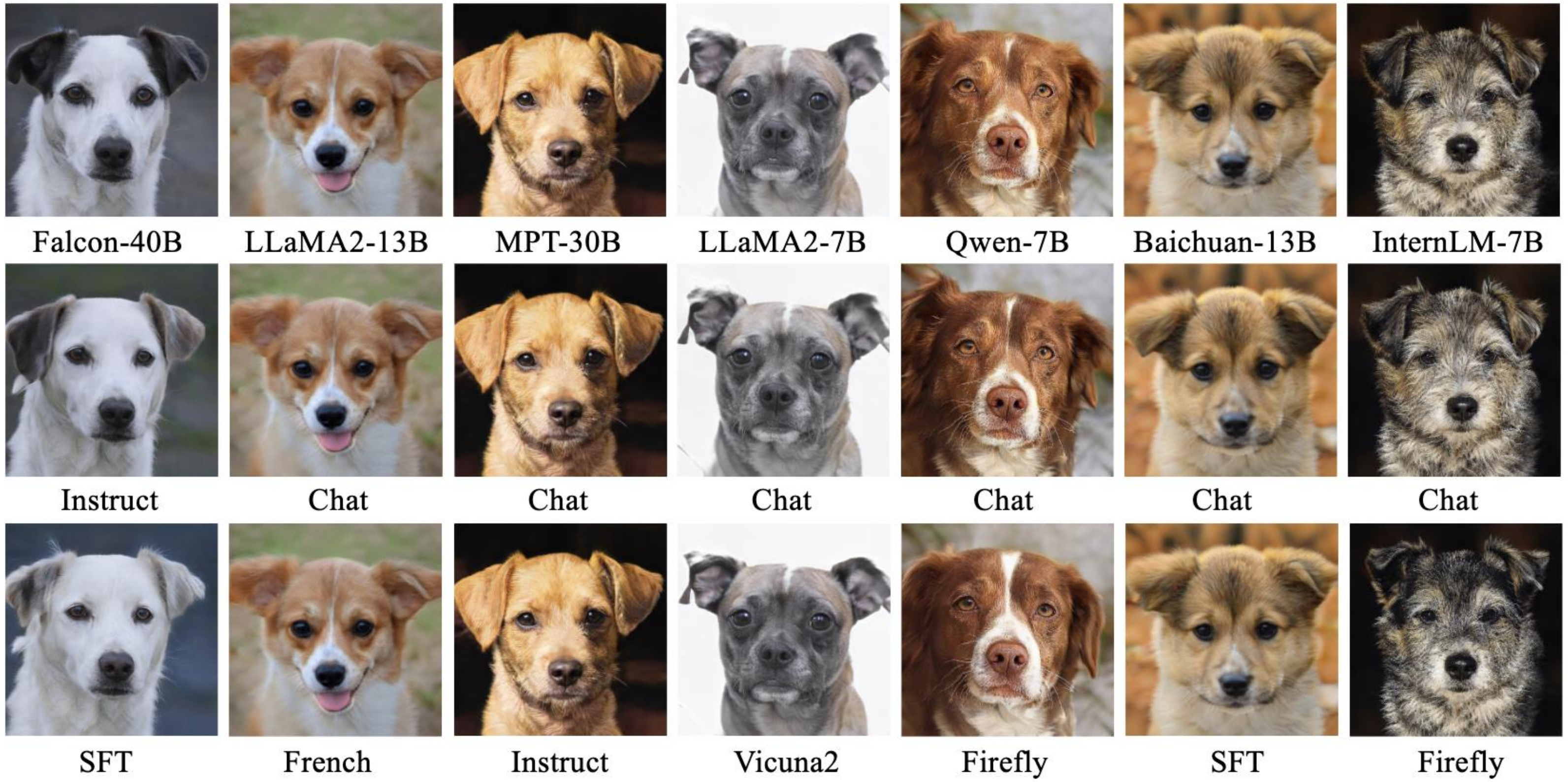


Experiments

1. 7 Independently Trained LLMs and Their Offspring Models
2. LLaMA family models
3. 28 independently trained LLMs.
4. Quantitatively evaluate the discrimination ability of the fingerprints through human subject study.

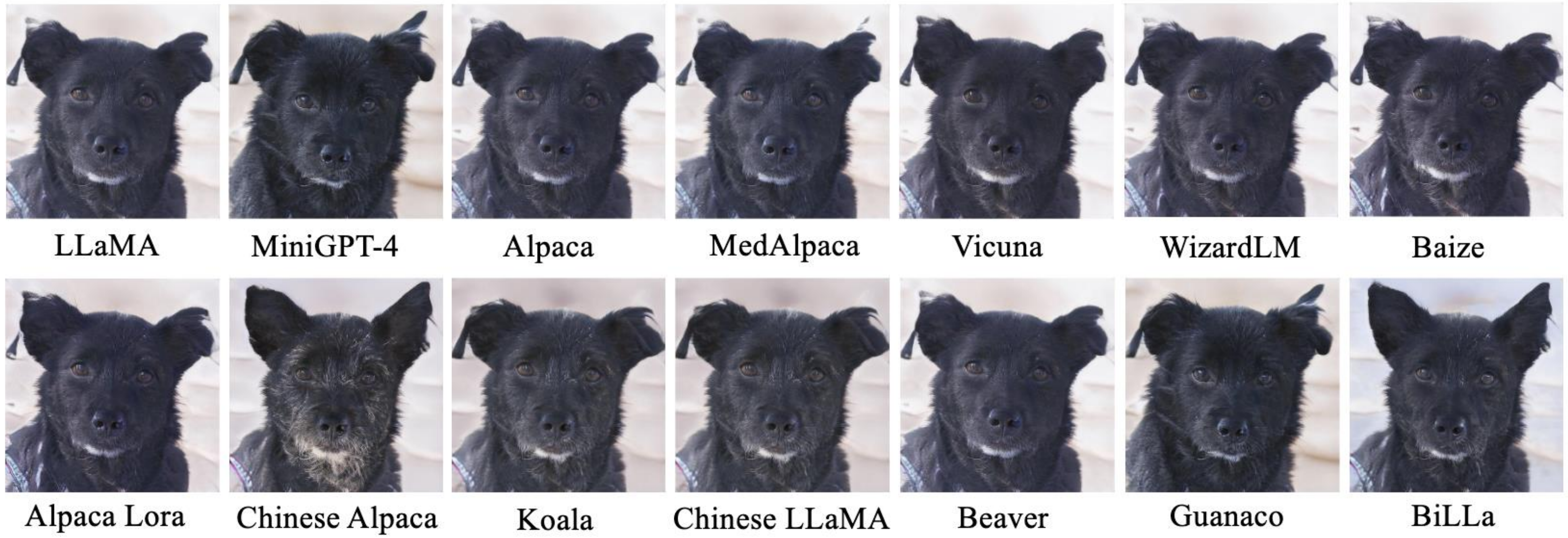
Independently Trained LLMs and Their Offspring Models

ICS	Falcon-40B	LLaMA2-13B	MPT-30B	LLaMA2-7B	Qwen-7B	Baichuan-13B	InternLM-7B
Offspring1	99.61	99.50	99.99	99.47	98.98	99.76	99.28
Offspring2	99.69	99.49	99.99	99.41	99.71	99.98	99.02



LLaMA family models

ICS	LLaMA	MiGPT	Alpaca	MAlpaca	Vicuna	Wizard	Baize	AlpacaL	CAlpaca	Koala	CLLaMA	Beaver	Guanaco	BiLLa
LLaMA	100.00	99.20	99.95	99.86	99.42	99.89	99.60	99.60	91.35	99.63	93.57	99.97	92.62	82.56
MiGPT	99.20	100.00	99.17	99.10	99.10	99.15	98.83	98.82	90.65	99.00	92.84	99.19	91.93	82.24
Alpaca	99.95	99.17	100.00	99.82	99.38	99.85	99.55	99.57	91.31	99.59	93.53	99.97	92.59	82.52
MAlpaca	99.86	99.10	99.82	100.00	99.31	99.76	99.46	99.47	91.23	99.51	93.45	99.84	92.50	82.51
Vicuna	99.42	99.10	99.38	99.31	100.00	99.35	99.05	99.04	90.84	99.15	93.04	99.41	92.14	82.28
Wizard	99.89	99.15	99.85	99.76	99.35	100.00	99.50	99.50	91.25	99.56	93.47	99.87	92.52	82.57
Baize	99.60	98.83	99.55	99.46	99.05	99.50	100.00	99.23	90.97	99.25	93.19	99.57	92.25	82.25
AlpacaL	99.60	98.82	99.57	99.47	99.04	99.50	99.23	100.00	90.99	99.24	93.21	99.59	92.31	82.30
CAlpaca	91.35	90.65	91.31	91.23	90.84	91.25	90.97	90.99	100.00	91.04	97.44	91.33	85.19	75.60
Koala	99.63	99.00	99.59	99.51	99.15	99.56	99.25	99.24	91.04	100.00	93.23	99.61	92.27	82.34
CLLaMA	93.57	92.84	93.53	93.45	93.04	93.47	93.19	93.21	97.44	93.23	100.00	93.55	86.80	77.41
Beaver	99.97	99.19	99.97	99.84	99.41	99.87	99.57	99.59	91.33	99.61	93.55	100.00	92.60	82.57
Guanaco	92.62	91.93	92.59	92.50	92.14	92.52	92.25	92.31	85.19	92.27	86.80	92.60	100.00	77.17
BiLLa	82.56	82.24	82.52	82.51	82.28	82.57	82.25	82.30	75.60	82.34	77.41	82.57	77.17	100.00

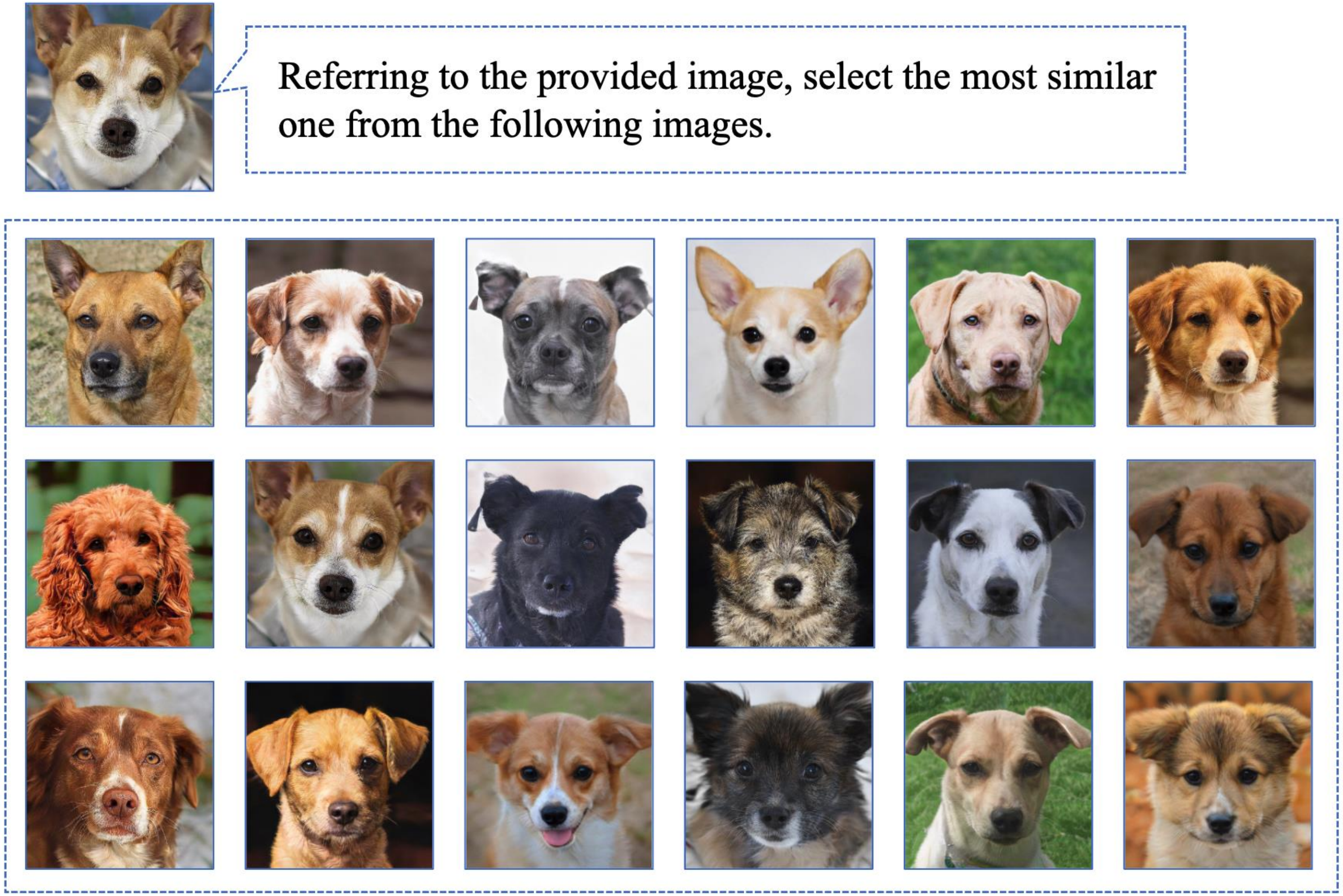


Fingerprints of 28 independently trained LLMs.

ICS between 28 independently trained LLMs

ICS	GPT2	CGPT	CLM	CLM2	OPT6.7	Py6.9	MPT7	Bai7	Fal7	Inte7	OLM	LM7	Qw7	Bloom	LM27	RedP	Py12	LM213	Bai13	LM13	Neox	LM30	OPT30	Fal40	LM65	Qw72	Gal120	Fal180
GPT2	100.00	18.06	-0.67	0.01	5.50	0.03	0.53	0.16	0.30	0.21	0.05	-0.15	-0.07	-0.45	-0.04	0.03	-0.04	0.09	0.30	-0.09	0.11	-0.09	3.36	0.79	-0.24	-0.09	-0.37	-1.35
CGPT	18.06	100.00	-0.29	0.08	7.46	0.14	1.06	0.23	0.48	0.07	0.23	-0.30	0.10	-0.79	-0.18	0.74	0.04	0.01	0.25	0.02	0.05	-0.08	5.10	0.17	0.03	-0.18	-0.18	-1.07
CLM	-0.67	-0.29	100.00	0.18	-1.07	-0.01	-1.32	-0.14	-0.09	-0.18	-0.09	0.15	0.14	0.37	-0.12	0.04	0.10	0.28	-0.03	-0.01	-0.10	-0.07	-0.73	0.17	0.18	0.05	-1.04	0.27
CLM2	0.01	0.08	0.18	100.00	-0.05	0.75	-0.08	0.11	0.87	0.24	0.11	0.11	0.14	0.79	0.10	0.92	0.69	0.14	0.11	0.09	0.60	-0.02	-0.07	0.30	-0.03	0.07	-0.01	-0.03
OPT6.7	5.50	7.46	-1.07	-0.05	100.00	0.45	5.87	0.41	0.48	-0.06	-0.06	-0.36	-0.14	-1.09	0.02	1.31	0.17	-0.13	0.17	-0.23	0.15	-0.38	46.29	0.65	-0.03	-0.11	-0.17	-1.26
Py6.9	0.03	0.14	-0.01	0.75	0.45	100.00	0.13	0.01	0.66	-0.06	0.04	-0.00	0.01	0.55	0.02	2.37	1.58	0.02	0.01	-0.02	1.41	-0.00	0.29	0.23	-0.01	0.02	-0.01	-0.04
MPT7	0.53	1.06	-1.32	-0.08	5.87	0.13	100.00	0.32	0.44	0.13	0.10	-0.13	-0.12	0.83	-0.10	0.62	0.40	-0.18	0.52	-0.03	-0.28	-0.49	1.10	-1.23	-0.33	-0.12	-0.61	-0.82
Bai7	0.16	0.23	-0.14	0.11	0.41	0.01	0.32	100.00	0.13	0.21	0.21	0.32	0.41	-0.13	0.35	0.09	0.00	0.22	0.42	0.28	0.04	0.10	0.21	-0.08	0.10	0.31	-0.16	0.01
Fal7	0.30	0.48	-0.09	0.87	0.48	0.66	0.44	0.13	100.00	-0.06	0.04	0.08	0.13	0.48	0.23	0.84	0.62	0.05	0.16	0.01	0.54	0.19	0.39	1.68	0.05	0.19	0.01	-11.07
Inte7	0.21	0.07	-0.18	0.24	-0.06	-0.06	0.13	0.21	-0.06	100.00	0.18	0.03	0.48	-0.01	-0.13	0.02	0.02	0.36	0.13	0.08	-0.00	-0.31	-0.64	0.08	-0.29	-0.26	0.00	-0.01
OLM	0.05	0.23	-0.09	0.11	-0.06	0.04	0.10	0.21	0.04	0.18	100.00	0.32	0.32	0.09	0.39	0.06	0.03	0.23	0.35	0.27	0.05	0.19	0.01	-0.04	0.06	0.32	0.08	-0.06
LM7	-0.15	-0.30	0.15	0.11	-0.36	-0.00	-0.13	0.32	0.08	0.03	0.32	100.00	0.60	0.08	3.16	0.06	0.02	1.64	0.62	2.07	0.00	1.15	0.04	-0.02	1.59	0.67	0.06	0.04
Qw7	-0.07	0.10	0.14	0.14	-0.14	0.01	-0.12	0.41	0.13	0.48	0.32	0.60	100.00	0.01	0.53	-0.02	0.04	0.46	0.57	0.42	-0.00	-0.20	-0.12	-0.08	0.03	0.76	0.11	-0.01
Bloom	-0.45	-0.79	0.37	0.79	-1.09	0.55	0.83	-0.13	0.48	-0.01	0.09	0.08	0.01	100.00	-0.09	0.35	0.48	0.11	0.03	0.07	0.41	0.02	-0.68	0.05	-0.08	0.01	-0.00	-0.18
LM27	-0.04	-0.18	-0.12	0.10	0.02	0.02	-0.10	0.35	0.23	-0.13	0.39	3.16	0.53	-0.09	100.00	-0.04	-0.03	1.45	0.64	1.67	0.02	1.77	0.37	-0.04	1.71	0.87	0.15	0.16
RedP	0.03	0.74	0.04	0.92	1.31	2.37	0.62	0.09	0.84	0.02	0.06	0.06	-0.02	0.35	-0.04	100.00	2.08	-0.00	-0.02	0.03	1.91	-0.13	0.68	0.29	0.03	0.12	0.21	-0.15
Py12	-0.04	0.04	0.10	0.69	0.17	1.58	0.40	0.00	0.62	0.02	0.03	0.02	0.04	0.48	-0.03	2.08	100.00	0.04	-0.01	-0.02	1.27	-0.02	0.08	0.30	-0.04	-0.03	-0.03	-0.00
LM213	0.09	0.01	0.28	0.14	-0.13	0.02	-0.18	0.22	0.05	0.36	0.23	1.64	0.46	0.11	1.45	-0.00	0.04	100.00	0.35	1.03	-0.01	-0.06	-0.39	-0.00	0.15	0.20	-0.06	0.13
Bai13	0.30	0.25	-0.03	0.11	0.17	0.01	0.52	0.42	0.16	0.13	0.35	0.62	0.57	0.03	0.64	-0.02	-0.01	0.35	100.00	0.41	-0.01	0.21	0.21	-0.14	0.25	0.59	0.02	-0.10
LM13	-0.09	0.02	-0.01	0.09	-0.23	-0.02	-0.03	0.28	0.01	0.08	0.27	2.07	0.42	0.07	1.67	0.03	-0.02	1.03	0.41	100.00	-0.01	0.39	0.13	-0.12	0.88	0.37	0.07	-0.04
Neox	0.11	0.05	-0.10	0.60	0.15	1.41	-0.28	0.04	0.54	-0.00	0.05	0.00	-0.00	0.41	0.02	1.91	1.27	-0.01	-0.01	-0.01	100.00	-0.00	0.14	0.34	0.02	0.03	0.11	0.01
LM30	-0.09	-0.08	-0.07	-0.02	-0.38	-0.00	-0.49	0.10	0.19	-0.31	0.19	1.15	-0.20	0.02	1.77	-0.13	-0.02	-0.06	0.21	0.39	-0.00	100.00	0.12	0.08	2.45	0.48	-0.13	0.06
OPT30	3.36	5.10	-0.73	-0.07	46.29	0.29	1.10	0.21	0.39	-0.64	0.01	0.04	-0.12	-0.68	0.37	0.68	0.08	-0.39	0.21	0.13	0.14	0.12	100.00	0.55	0.56	0.40	-0.06	-0.93
Fal40	0.79	0.17	0.17	0.30	0.65	0.23	-1.23	-0.08	1.68	0.08	-0.04	-0.02	-0.08	0.05	-0.04	0.29	0.30	-0.00	-0.14	-0.12	0.34	0.08	0.55	100.00	-0.05	-0.10	0.20	4.90
LM65	-0.24	0.03	0.18	-0.03	-0.03	-0.01	-0.33	0.10	0.05	-0.29	0.06	1.59	0.03	-0.08	1.71	0.03	-0.04	0.15	0.25	0.88	0.02	2.45	0.56	-0.05	100.00	0.44	-0.13	0.02
Qw72	-0.09	-0.18	0.05	0.07	-0.11	0.02	-0.12	0.31	0.19	-0.26	0.32	0.67	0.76	0.01	0.87	0.12	-0.03	0.20	0.59	0.37	0.03	0.48	0.40	-0.10	0.44	100.00	0.07	0.09
Gal120	-0.37	-0.18	-1.04	-0.01	-0.17	-0.01	-0.61	-0.16	0.01	0.00	0.08	0.06	0.11	-0.00	0.15	0.21	-0.03	-0.06	0.02	0.07	0.11	-0.13	-0.06	0.20	-0.13	0.07	100.00	0.19
Fal180	-1.35	-1.07	0.27	-0.03	-1.26	-0.04	-0.82	0.01	-11.07	-0.01	-0.06	0.04	-0.01	-0.18	0.16	-0.15	-0.00	0.13	-0.10	-0.04	0.01	0.06	-0.93	4.90	0.02	0.09	0.19	100.00

Human subject study



Yielded a **94.74%** accuracy rate among 72 college-educated individuals, each answering 51 questions.

Limitations

1. Our focus is solely on transformer-based LLMs, and generalizing our approach to other architectures requires further investigation

2. StyleGAN2's behavior exhibits occasional inconsistencies, leading to the generation of similar images for dissimilar models or dissimilar images for highly similar models.

Thank you!