

Espresso

Robust Concept Filtering in Text-to-Image Models

Anudeep Das¹ Vasisht Duddu¹ Rui Zhang² N. Asokan¹
¹University of Waterloo ²Zhejiang University

BEST PAPER AWARD
ACM CODASPY 2025 · Pittsburgh, PA

ssg-research.github.io/mlsec/conceptfilter



Paper · code · talk

Key idea: filter each generated image by comparing its CLIP embedding to the text embeddings of **both** the unacceptable concept **and** an acceptable one — then fine-tune CLIP to **push the two concepts apart**. The T2I model is never modified.

0.06 vs 0.24

R1 Effective: mean CLIP accuracy on unacceptable prompts (best baseline: UCE)

0.04 vs 0.29

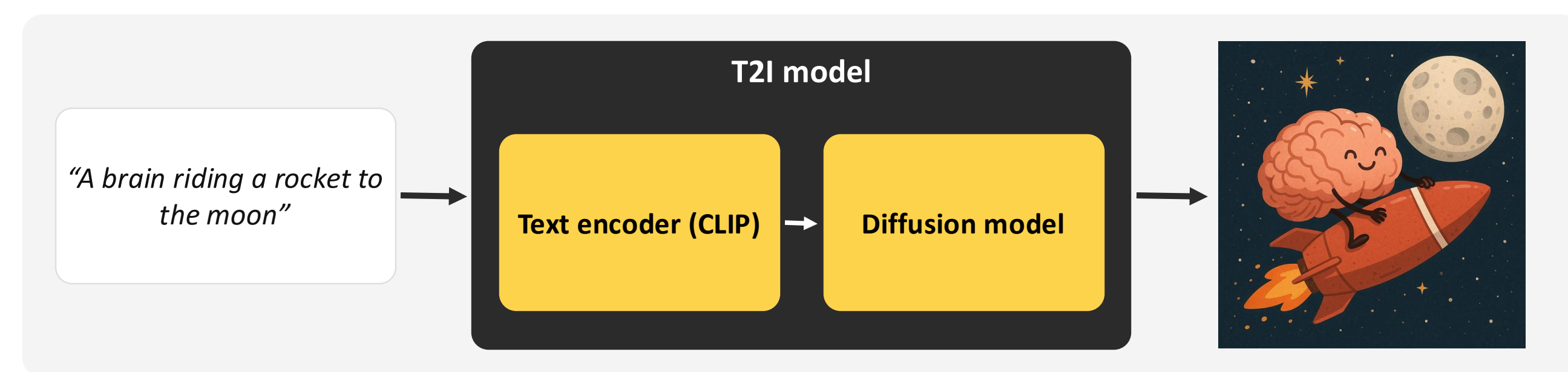
R3 Robust: mean CLIP accuracy under white-box CCE+ attack (best baseline: Mod)

9.1 min

Efficient: to fine-tune on one A100 — vs 60–135 min for CA, SA, ESD, SDD, Mod

1 The Problem

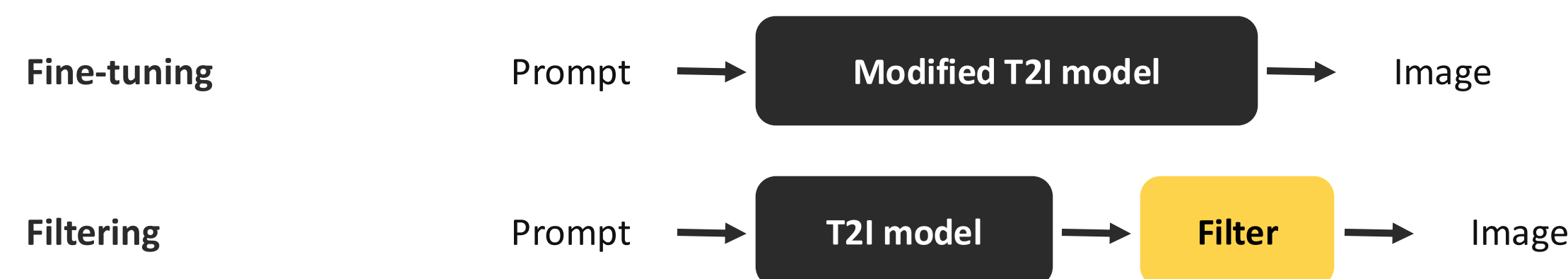
Text-to-image (T2I) models are trained on huge, **unfiltered Internet data** and can generate **unacceptable concepts**.



Three groups of unacceptable concepts studied

Inappropriate nudity · violence · disturbing · hateful	Copyright-infringing Grumpy Cat · Nemo · Captain Marvel · Snoopy · R2D2	Unauthorized likeness Taylor Swift · Angelina Jolie · Brad Pitt · Elon Musk
--	---	---

2 Concept Removal Techniques



An ideal CRT must be...

- R1 Effective** — minimize generation of unacceptable images
- R2 Utility-preserving** — keep quality of acceptable images (and don't block them)
- R3 Robust** — resist adversarial prompts that force unacceptable images

Gap: no prior CRT meets all three. Fine-tuning CRTs trade utility for effectiveness and ignore robustness; filters (e.g., UD) are easily evaded.

3 Threat Model & Evaluation

Adversary goal: craft prompts that evade the CRT yet still yield unacceptable images.

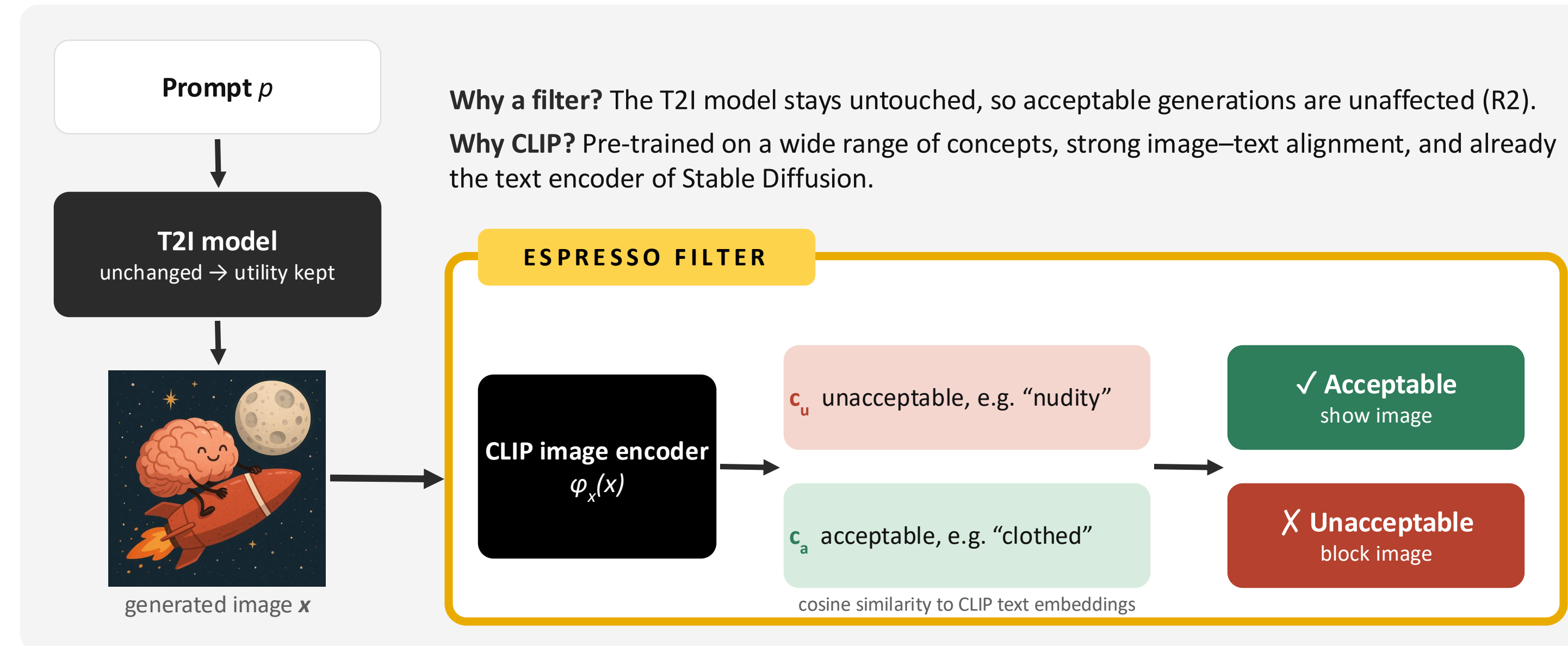
Worst case: white-box access to a local copy of the T2I model and filter.

CRT-aware attacks: we extend PEZ, Typo, RingBell and CCE to also move toward the acceptable concept (PEZ+, Typo+, RingBell+, CCE+); plus SneakyPrompt.

Baselines: 7 fine-tuning CRTs (CA, FMN, SA, ESD, UCE, SDD, Mod) and UD filter.

Metrics: R1/R3 — CLIP accuracy or false-negative rate on unacceptable prompts (↓); R2 — normalized CLIP score on COCO prompts (↑) or false-positive rate (↓).

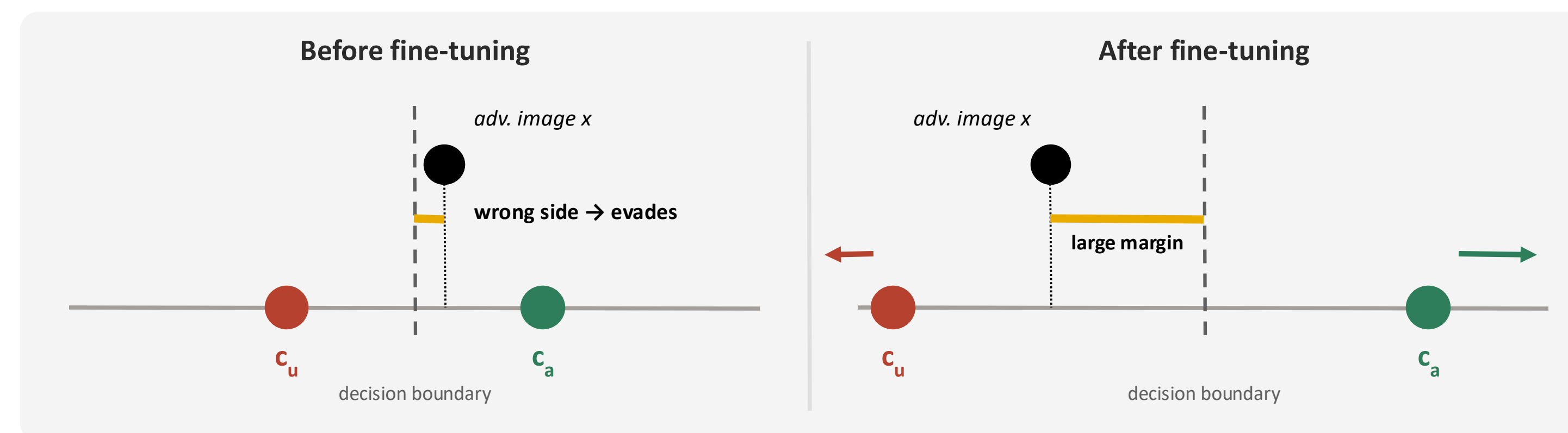
4 Espresso: How It Works



$$F(x) = \operatorname{argmax}_{i \in \{a,u\}} \exp(\cos(\varphi_x(x), \varphi_p(c_i)) / \tau) / \sum_j \exp(\cos(\varphi_x(x), \varphi_p(c_j)) / \tau)$$

temperature $\tau = 0.01$ · CLIP image encoder φ_x , text encoder φ_p

5 Fine-tune to Push Concepts Apart



Fine-tuning separates acceptable and unacceptable concepts in CLIP's embedding space, **increasing the distance to the decision boundary** — like adversarial training, adversarial prompts become hard to find. Group 2–3: maximize $\|\varphi_p(p^u) - \varphi_p(p^a)\|$; Group 1: contrastive image–concept losses.

6 Beats the UD Filter

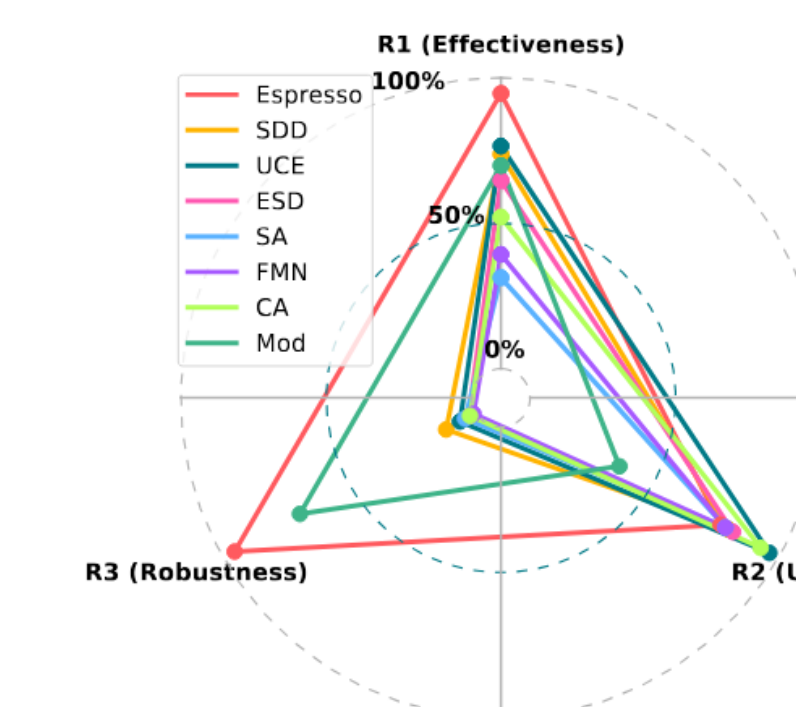
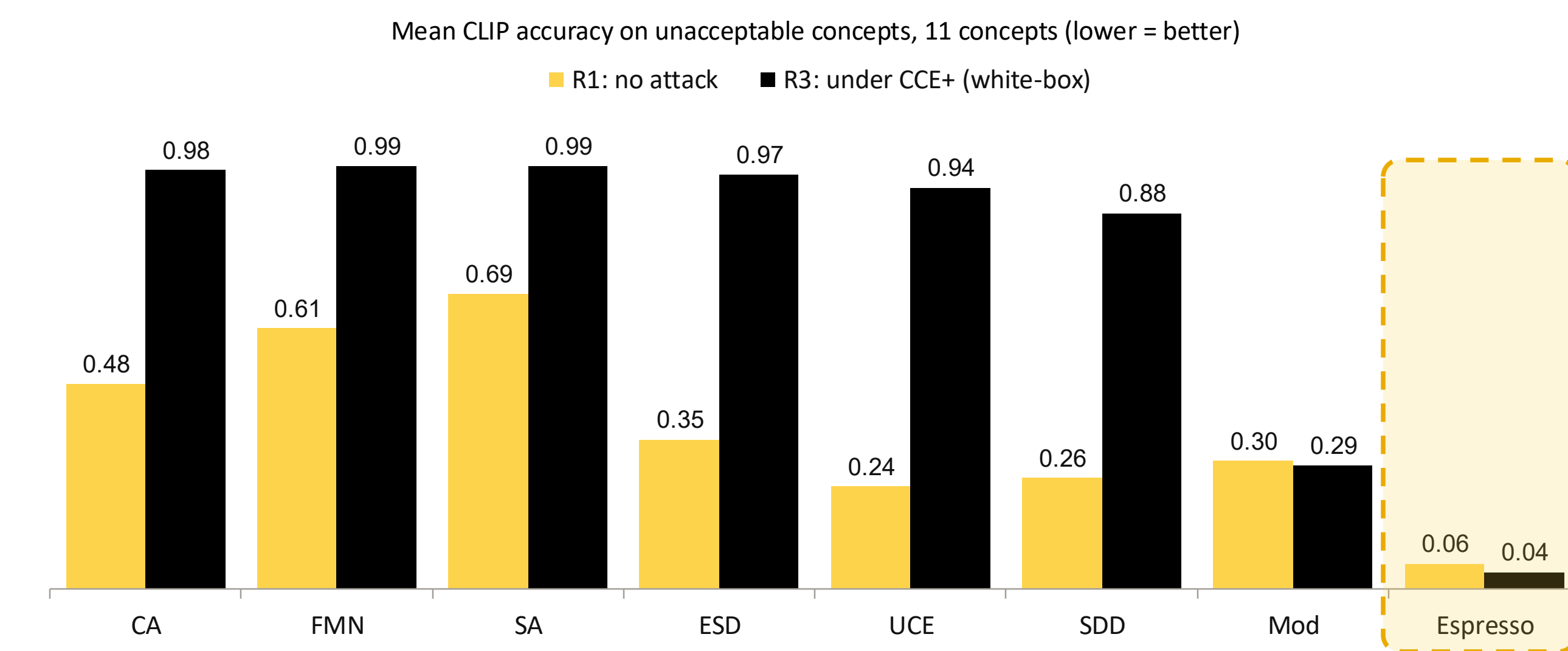
False-negative rate on unacceptable images (lower = better)

Attack	Nudity		Violence		Disturbing		Hateful	
	UD	Esp.	UD	Esp.	UD	Esp.	UD	Esp.
No attack	0.39	0.15	0.90	0.20	0.89	0.53	1.00	0.42
Typo+	0.55	0.15	0.91	0.26	0.39	0.39	0.48	0.37
PEZ+	0.65	0.16	0.91	0.25	0.89	0.14	1.00	0.20
CCE+	0.00	0.00	0.75	0.38	1.00	0.02	1.00	0.02
RingBell+	0.95	0.06	0.50	0.08	0.30	0.06	0.90	0.25
SneakyPrompt	0.67	0.40	0.71	0.14	0.82	0.70	0.48	0.15

< 0.25 0.25–0.50 > 0.50

Utility: both keep FPR ≤ 0.08

7 Beats 7 Fine-tuning CRTs



• **Most effective & robust** across concepts and all five attacks (exception: broad concept “violence”).

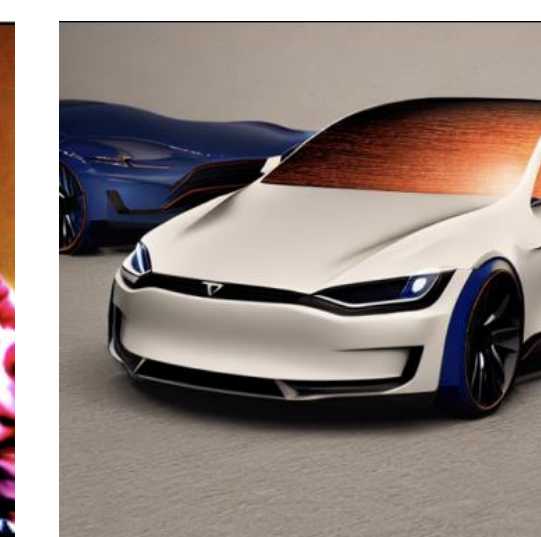
Utility: small drop (normalized CLIP score ≥ 0.94 on 6 of 11 concepts); ambiguous names like “Nemo” improve once made specific.

8 Why Is Espresso Robust?

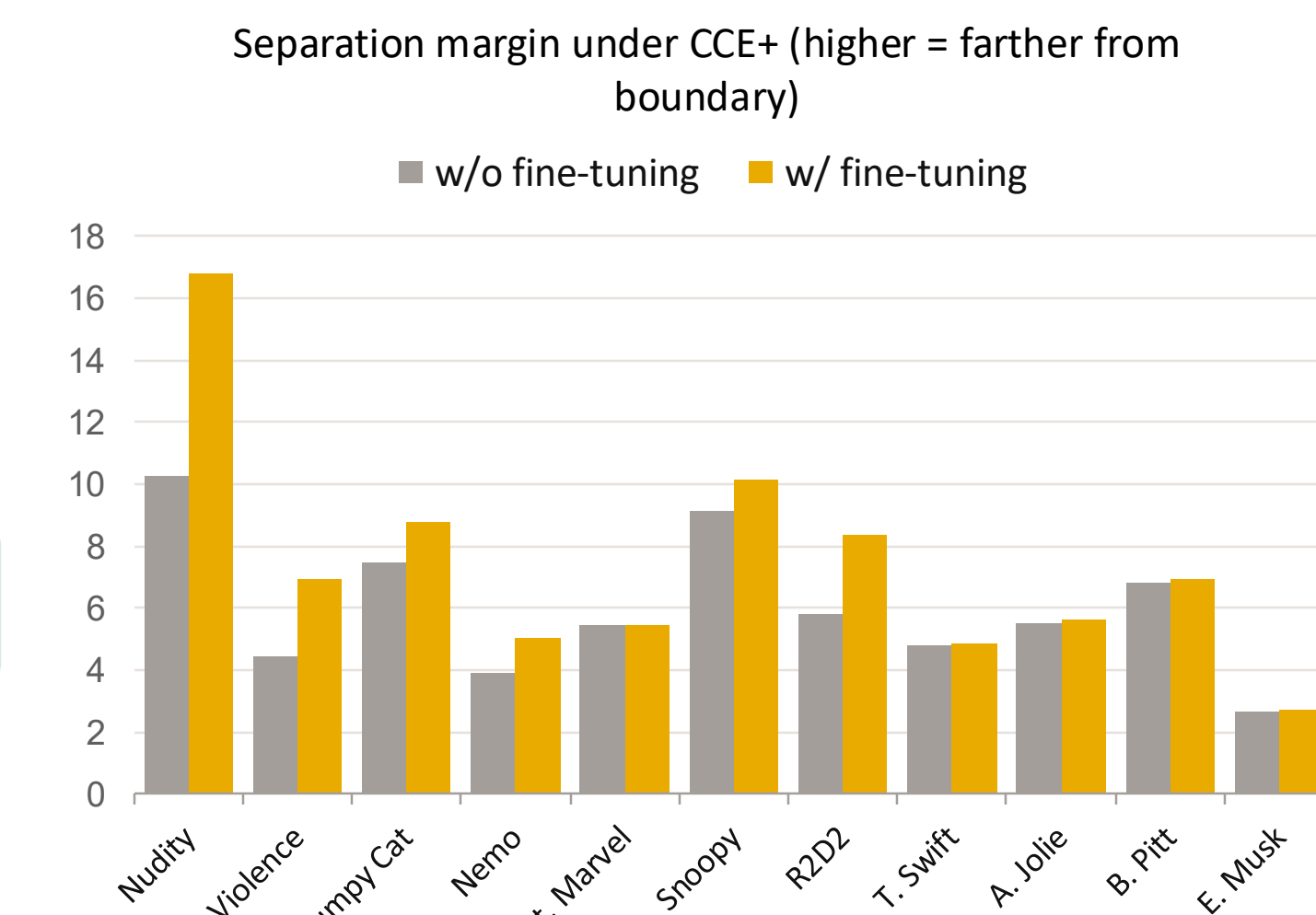
Attacks face a **dilemma**: keep the concept and get caught, or evade and lose it.



Elon Musk image
→ detected



Adversarial prompt
→ Musk lost



9 Take-aways

Espresso is the first robust concept filter: it checks each generated image against both the unacceptable and an acceptable concept.

Better trade-off than all 8 CRTs: more effective and more robust, with only a small utility drop and no change to the T2I model.

Cheap to deploy: 9.1 min of fine-tuning on one A100.