# Thematic session

Paper presentation: Group4

MultiX

# Nanne van Noord

Published as a conference paper at ICLR 2025

## ALIGNING VISUAL CONTRASTIVE LEARNING MODELS VIA PREFERENCE OPTIMIZATION

**Amirabbas Afzali & Borna Khodabandeh** *
{amir8afzali, borna710kh}@gmail.com

**Ali Rasekh**
L3S Research Center,
Leibniz Universität Hannover, Germany
ali.rasekh@L3S.de

**Mahyar JafariNodeh**
Massachusetts Institute of Technology, USA
mahyarjn@mit.edu

**Sepehr Kazemi**
sepehrkazemi9@gmail.com

**Simon Gottschalk**
L3S Research Center,
Leibniz Universität Hannover, Germany
gottschalk@L3S.de

# Preference Optimization (PO)

Reinforcement Learning from Human Feedback (RLHF):

- Annotators rank model responses, e.g., for input x: $y_l < y_w$
    - Resulting in a dataset: $D = \{(x, y_w, y_l)\}$
- Supervised train a reward model on dataset:
- RL train model:

$$(r_\phi) = \mathbb{E}_{x,y_w,y_l\sim\mathcal{D}}[-\log(\sigma(r_\phi(x,y_w) - r_\phi(x,y_l)))].$$

Slow & Unstable
(due to sampling and RL)

$$\max_{\pi_\theta} \mathbb{E}_{x\sim\mathcal{D},} \ldots (y|x) \| \pi_{\mathrm{ref}}(y|x)),$$

Direct Preference Optimization:

- DPO: Tends to overfit

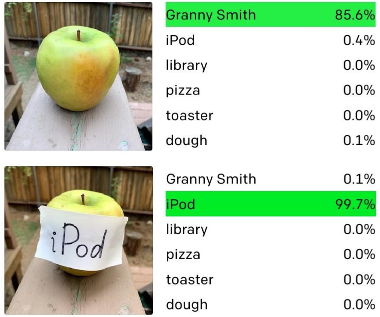
$$\mathcal{L}_{\mathrm{DPO}}(\pi_\theta; \pi_{\mathrm{ref}}) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\log\sigma\left(\beta\log\frac{\pi_\theta(y_w \mid x)}{\pi_{\mathrm{ref}}(y_w \mid x)} - \beta\log\frac{\pi_\theta(y_l \mid x)}{\pi_{\mathrm{ref}}(y_l \mid x)}\right)\right].$$

- Identity Preference Optimization (IPO): Adds regularisation term
- Kahneman-Tversky Optimization (KTO): does not need ranking, just binary of whether outcome is desired/undesired

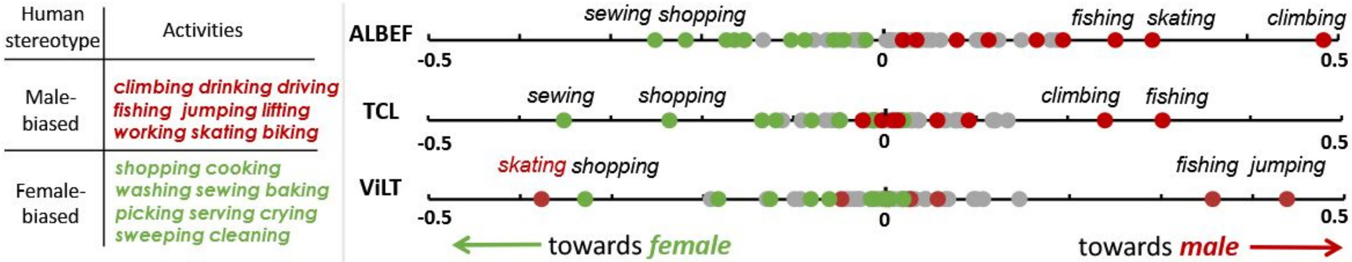
# Aligning Visual Contrastive Learning Models

PO mainly applied to generative models, this work focuses on contrastive visual (e.g., CLIP) for:

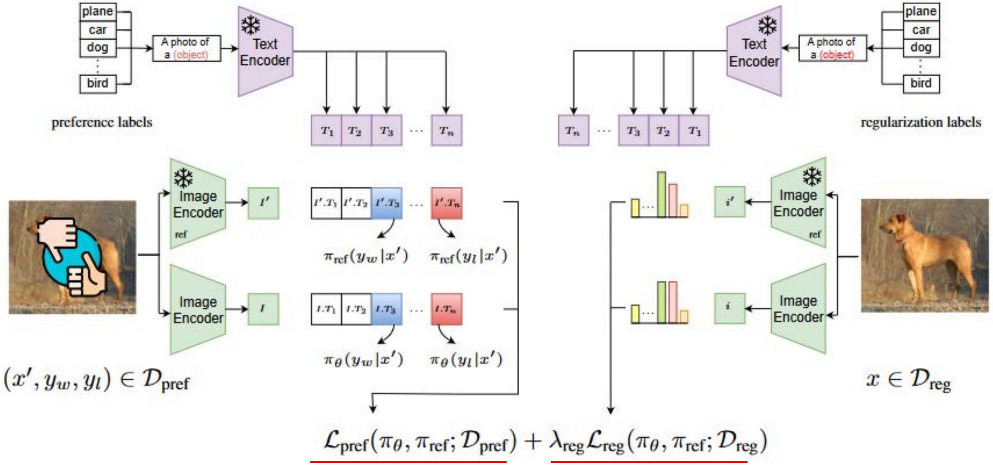
- Mitigating typographic attacks




- Mitigating gender bias



# Preference-based Contrastive Optimization




**Algorithm 1** Preference-based contrastive optimization

**Require:** dataset $\mathcal{D} = (\mathcal{D}_{\text{pref}}, \mathcal{D}_{\text{reg}})$, Model $\pi_\theta$, Reference model $\pi_{\text{ref}}$, Regularization coef. $\lambda_{\text{reg}}$

**for** each batch $b \in \mathcal{D}$ **do**

  $b_{\text{pref}}, b_{\text{reg}} \leftarrow b$ ▷ Get batch of preference / regularization data

  Compute $\pi_\Psi(y|x) \triangleq \text{Softmax}(f_\Psi(y, x))$ for $\Psi \in \{\theta, \text{ref}\}$ ▷ Compute model and ref. distributions

  $l_{\text{pref}} \leftarrow \mathcal{L}_{\text{po}}(\pi_\theta, \pi_{\text{ref}}; b_{\text{pref}})$ ▷ Compute preference loss using one of Eqs. (5), (6), or (7)

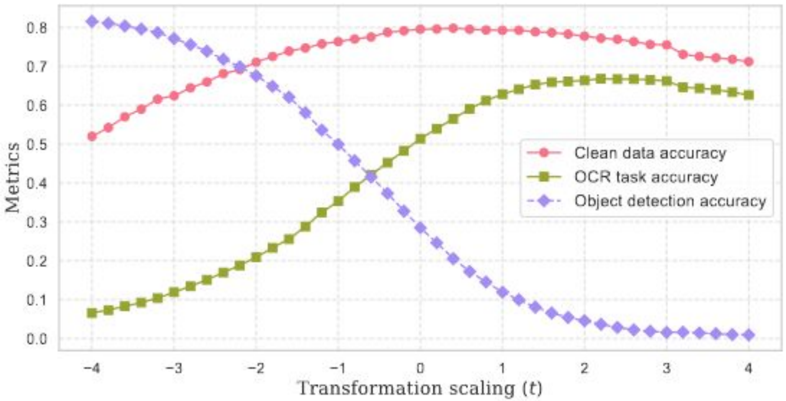
  $l_{\text{reg}} \leftarrow \mathcal{L}_{\text{reg}}(\pi_\theta, \pi_{\text{ref}}; b_{\text{reg}})$ ▷ Compute regularization loss as in Eq. (11), zero if $b_{\text{reg}} = \{\}$

  $l_{\text{tot}} \leftarrow l_{\text{pref}} + \lambda_{\text{reg}} \cdot l_{\text{reg}}$ ▷ Total loss

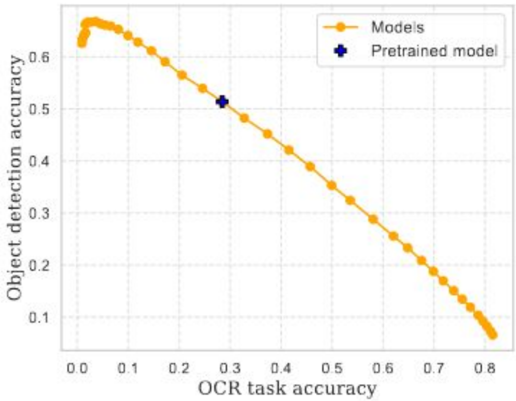
  Update model $\pi_\theta$ by minimizing $l_{\text{tot}}$

**end for**

# Results - Typographic Robustness




(a) Accuracy on typographic samples and percentage of typographic label predictions versus transformation scaling factor $t$. As $t$ increases, the model favors object labels over typographic labels while maintaining accuracy.




(b) Frontier of a DPO fine-tuned model, showing OCR vs. OD accuracy across with varying $t$.

Figure 2: Comparisons of optical character recognition (OCR) and object detection (OD).

# Results - Disentangling Gender Understanding

Figure 4: Images retrieved for the caption *"an image of a police"* with three different policies from top to bottom: reversed understanding of gender (6W, 4M), pretrained CLIP model (2W, 8M), neutralized understanding of gender (5W, 5M), i.e., $t = t^*$.

# Takeaways

- PO can be used on non-generative models
- Align without losing abilities (i.e., catastrophic forgetting)
- KTO performs best
    - easier to collect data for also!

Wangyuan Ding

# The Illusion of Thinking: Understanding the Strengths and Limitations of Reasoning Models via the Lens of Problem Complexity

Parshin Shojaee*† Iman Mirzadeh* Keivan Alizadeh
Maxwell Horton Samy Bengio Mehrdad Farajtabar

Apple

# Motivation

Large Reasoning Models (LRMs) show strong benchmark performance, but evaluations suffer from contamination and focus on final answers rather than reasoning paths.

The question is: can we find a way to show LRMs really doing reasoning?

Models to be tested: LRMs including Claude-3.7 Sonnet-Thinking, DeepSeek-R1, OpenAI o3-mini, and standard LLMs under identical compute budgets

Methods: use procedurally **generated puzzles** (Tower of Hanoi, River Crossing, Blocks World) with adjustable compositional complexity—ensuring zero overlap with training data—and analyze both final answers and **intermediate** reasoning traces

# Puzzles

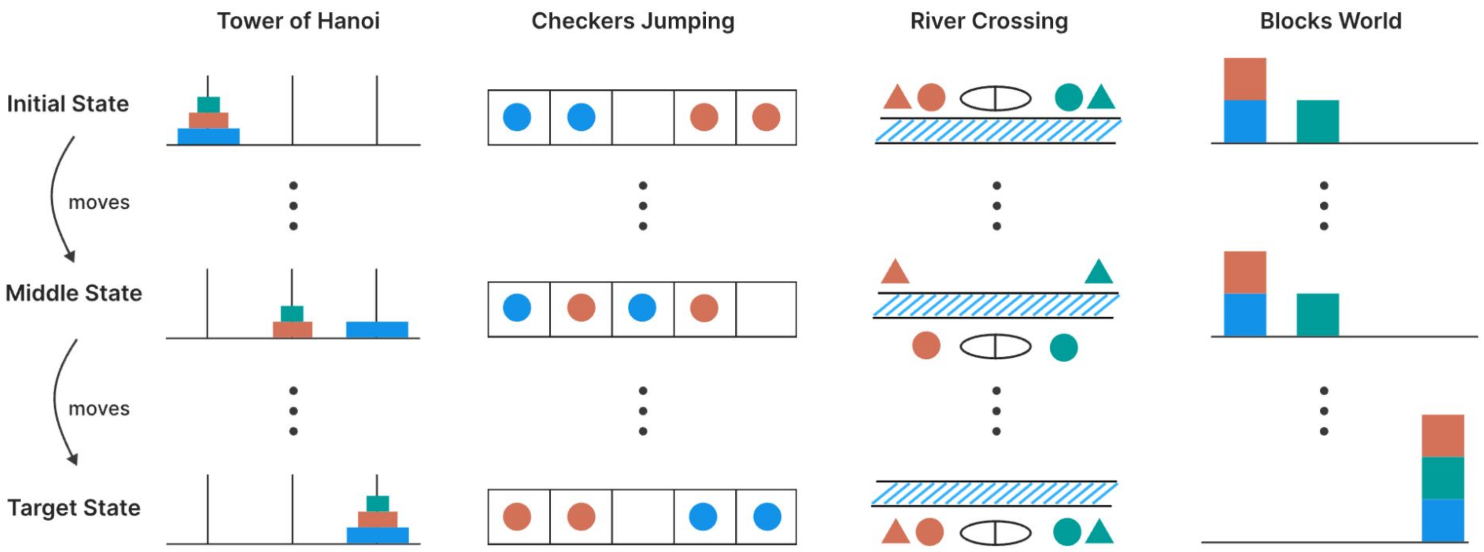



Figure 3: Illustration of the four puzzle environments. Columns show the progression from **initial state (top)** through **intermediate state (middle)** to **target state (bottom)** for puzzles: Tower of Hanoi (disk transfer across pegs), Checkers Jumping (position swapping of colored tokens), River Crossing (transporting entities across a river), and Blocks World (stack reconfiguration).

# Overall Structure

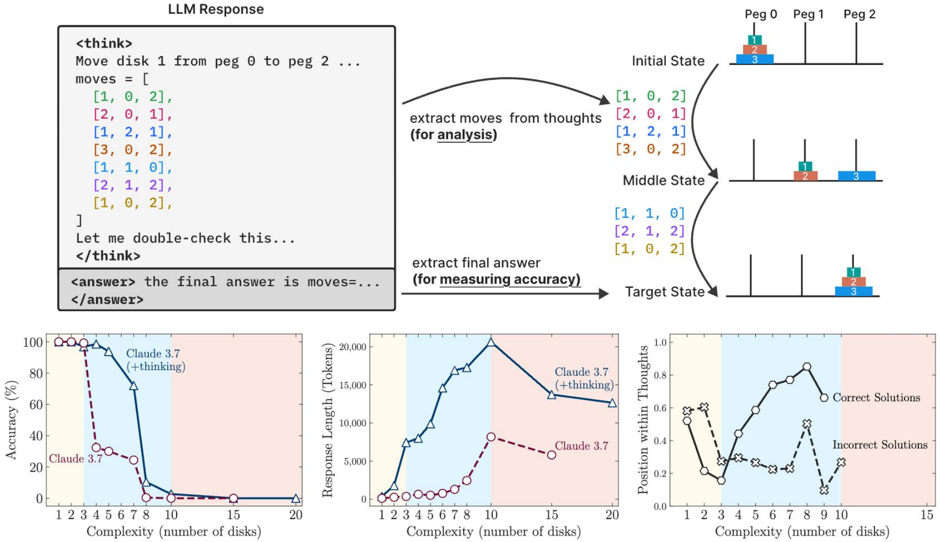



Figure 1: **Top**: Our setup enables verification of both final answers and intermediate reasoning traces, allowing detailed analysis of model thinking behavior. **Bottom left & middle**: At low complexity, non-thinking models are more accurate and token-efficient. As complexity increases, reasoning models outperform but require more tokens—until both collapse beyond a critical threshold, with shorter traces. **Bottom right**: For correctly solved cases, Claude 3.7 Thinking tends to find answers early at low complexity and later at higher complexity. In failed cases, it often fixates on an early wrong answer, wasting the remaining token budget. Both cases reveal inefficiencies in the reasoning process.

# Findings 1: Complexity Cliff

Performance drops sharply to near-zero at certain puzzle sizes—no gradual degradation—across all tested LRMs

1. Low complexity: Standard LLMs often outperform LRMs—LRMs may overthink.
2. Medium complexity: LRMs excel using their chain-of-thought.
3. High complexity: Both fail catastrophically—their “thinking” offers no rescue

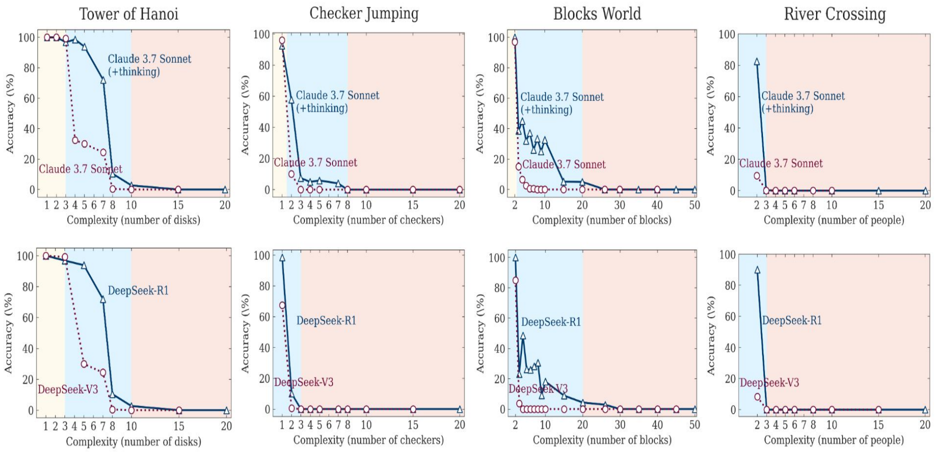



Figure 4: Accuracy of thinking models (Claude 3.7 Sonnet with thinking, DeepSeek-R1) versus their non-thinking counterparts (Claude 3.7 Sonnet, DeepSeek-V3) across all puzzle environments and varying levels of problem complexity.

# Findings 2: Failures

**Inference Paradox:** As complexity increases beyond a threshold, LRMs reduce token usage—they give up early despite remaining capacity—indicating a compute-scaling ceiling.

**Execution Failure:** Even if given the correct algorithm (e.g., Tower of Hanoi solution), LRMs don't improve—they still fail at the same complexity level.

**Inconsistent Reasoning:** A model might solve a 100-move problem but then fail a simpler 5-move variant, showing reasoning is brittle, not rule-based

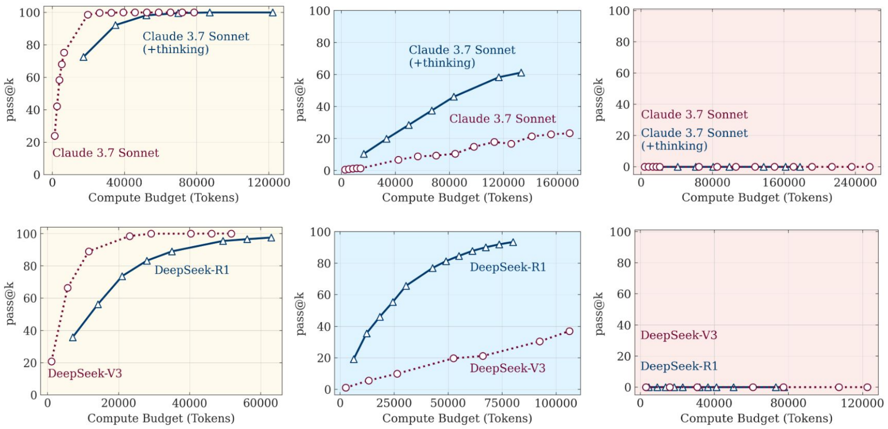



Figure 5: Pass@k performance of thinking vs. non-thinking models across equivalent compute budgets in puzzle environments of low, medium, and high complexity. Non-thinking models excel in simple problems, thinking models show advantages at medium complexity, while both approaches fail at high complexity regardless of compute allocation.

# Take-aways

**“Illusion” vs. genuine reasoning:** LRMs exhibit many outward signs of reasoning, but under controlled tests they fall apart—revealing pattern-matching, not structural thinking.

**Architectural limits:** Size and chain-of-thought alone don’t guarantee scalable reasoning—execution and generalization remain bottlenecks.

**Toward better evaluation:** Encourages flow-track analysis (reasoning traces) and procedurally validated benchmarks, not just outcome-based metrics.

**Future direction:** True reasoning may require new architectures—symbolic modules, grounding, memory systems, or synthesis of code—beyond next-token prediction.

**Thinking:** Do you think puzzles like Tower of Hanoi reflect general reasoning?

- “Tower of Hanoi… model decides that there’s too many steps… so it spins around trying to find a shortcut and fails.”
- “Breaking down after a few hundred reasoning steps doesn’t mean you’re not ‘really’ reasoning.”

# Carlo Bretti

**Video, How Do Your Tokens Merge?**

Sam Pollard
University of Bristol
sam.pollard@bristol.ac.uk

Michael Wray
University of Bristol
michael.wray@bristol.ac.uk

# Too many tokens

For a 5 minute video at 224 × 224 pixels consists of the typical 16 × 16 spatial patches, when sampling 1 frame per second, the transformer sequence length is 58,800 tokens.

Two main strategies to counter this: **dropout** and **merging**.

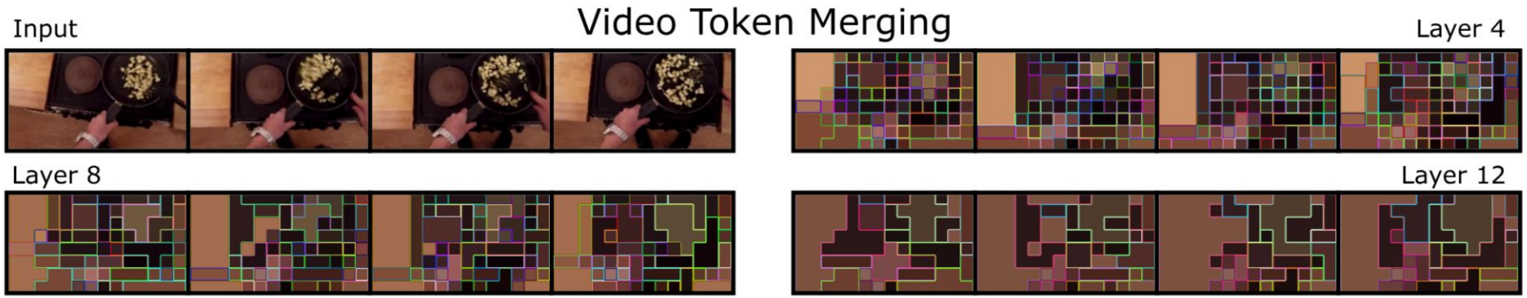



Figure 1. Video Token Merging reduces computation of video transformer models by successively merging tokens without re-training or additional learned parameters. We show how an input video has its tokens merged across different layers.

# Merging for Spatio-Temporal models

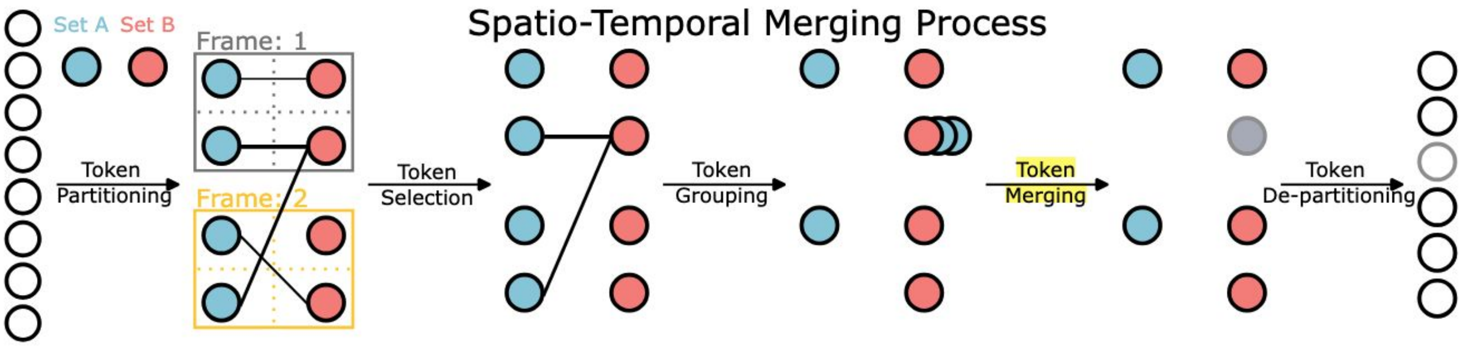
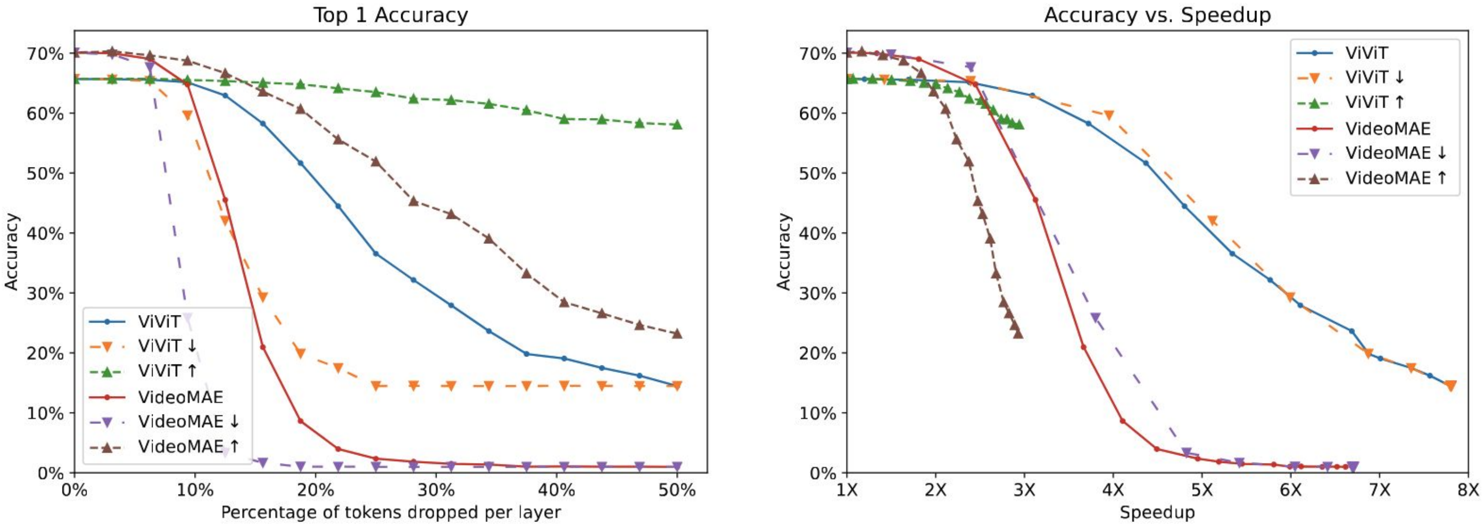



Figure 2. The merging process first separates tokens into two sets. Similarities are calculated and a one-to-many bipartite matching between tokens in each set is found. Finally, the top $r$ edges are kept and these are merged based on the similarity between tokens.

# Results on Kinetics (coarse video classification)




(a) Accuracy as $r$ increases.

(b) Accuracy against speedup

Figure 3. (Left) curves corresponds to accuracy with ViViT and VideoMAE on K400 when increasing $r$ (the number of tokens merged) up to its limit. The $x$-axis is the percentage (relative to the original total) of tokens dropped *per layer*. (Right) figure displays the accuracy against speedup gained for these $r$ values.

# Less gains on fine-grained data

Divided space-time models

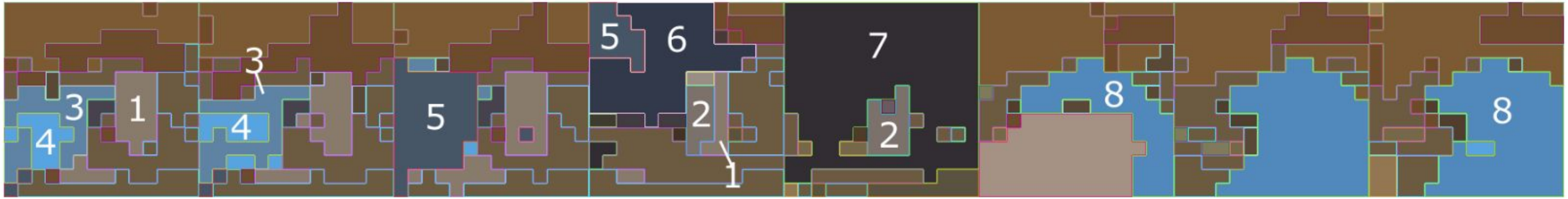
Spatio-temporal models

| Model | $r$ | Reduction | K400 | SSv2 | EK-100 Action | EK-100 Verb | EK-100 Noun | FPS | Speedup (X) |
|---|---|---|---|---|---|---|---|---|---|
| TimeSformer [2] | 0 | - | 76.63 | 50.66 | 31.32 | 55.48 | 47.23 | 117.78 | 1.00 |
| | $18 \times 8$ | random drop | 65.58 | 17.18 | 12.78 | 34.03 | 28.31 | 240.13 | 2.04 |
| | | drop | 68.30 | 22.97 | 16.09 | 38.24 | 33.02 | 240.13 | 2.04 |
| | | random merge | 38.41 | 5.46 | 2.82 | 21.47 | 8.57 | 234.58 | 1.99 |
| | | merge | **71.14** | **25.11** | **18.59** | **40.47** | **35.73** | 240.16 | 2.04 |
| Motionformer [26] | 0 | - | 70.50 | 61.39 | 35.02 | 61.09 | 46.72 | 99.79 | 1.00 |
| | $18 \times 8$ | random drop | 63.80 | **24.50** | 14.27 | 37.38 | 27.57 | 218.40 | 2.19 |
| | | drop | 63.41 | 22.46 | **15.92** | **39.54** | **30.60** | 216.73 | 2.17 |
| | | random merge | 49.30 | 17.76 | 7.88 | 31.56 | 16.77 | 210.30 | 2.11 |
| | | merge | **65.05** | 24.10 | 15.60 | 38.20 | 30.15 | 218.11 | 2.19 |
| VideoMAE [30] | 0 | - | 62.09 | 64.58 | 35.70 | 61.49 | 46.89 | 186.72 | 1.00 |
| | 150 | random drop | 55.02 | 57.29 | 28.53 | 55.45 | 39.45 | 481.45 | 2.58 |
| | | drop | **56.65** | 60.33 | 31.02 | 57.70 | **42.43** | 483.04 | 2.59 |
| | | random merge | 20.64 | 22.89 | 5.44 | 26.86 | 10.07 | 471.74 | 2.53 |
| | | merge | 56.10 | **61.10** | **31.27** | **58.00** | 42.39 | 476.28 | 2.55 |
| ViViT [1] | 0 | - | 63.43 | 50.63 | 35.82 | 58.19 | 51.59 | 106.00 | 1.00 |
| | 300 | random drop | 59.95 | 46.71 | 30.36 | 54.24 | 45.51 | 262.04 | 2.47 |
| | | drop | 58.00 | 45.36 | 30.12 | 53.20 | 46.90 | 262.34 | 2.47 |
| | | random merge | 28.88 | 19.15 | 5.78 | 28.88 | 14.82 | 259.92 | 2.45 |
| | | merge | **63.08** | **50.15** | **35.11** | **57.24** | **51.33** | 260.72 | 2.46 |

Table 1. Performance of token merging with a constant schedule when compared to alternative methods of reducing token sequence length. Bold indicates the reduction methods that achieve highest accuracy on a given dataset. Grey rows correspond to the upper bound accuracy of the original model.

# Qualitative results on fine-grained data

(a) Original clip.



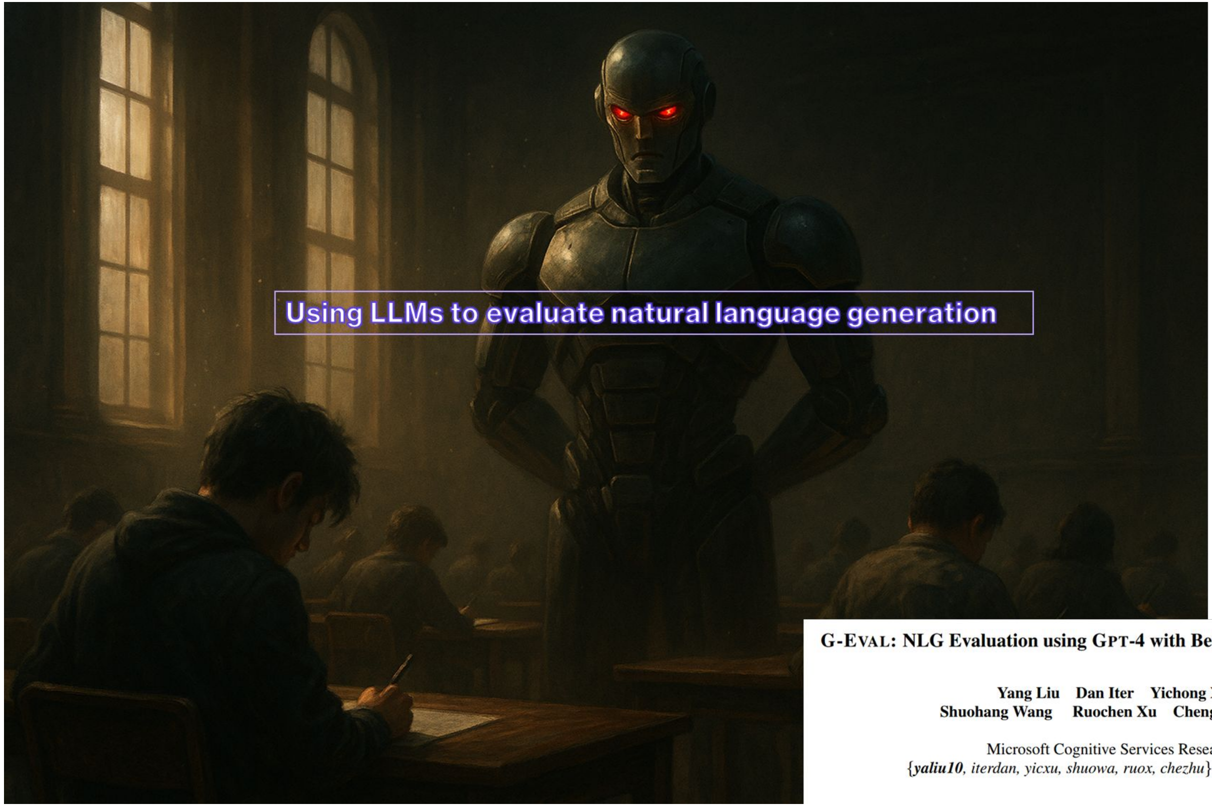

(b) Merged clip.

Figure 5. Visualisation of the final merged tokens for an SSv2 clip of "covering salt shaker with a towel", produced with VideoMAE. Tokens 1 and 2 capture the white salt shaker. The model struggles more with the blue towel, with it splitting into tokens 3 – 8.

# Takeaways

- Token merging can be an effective way to increase inference speed and works better than dropout
- Token merging is more damaging for fine-grained and temporal data (EK100 and SSv2)
- Might be useful for large scale processing and when you don’t need super fine grained information?

# Floris Gisolf

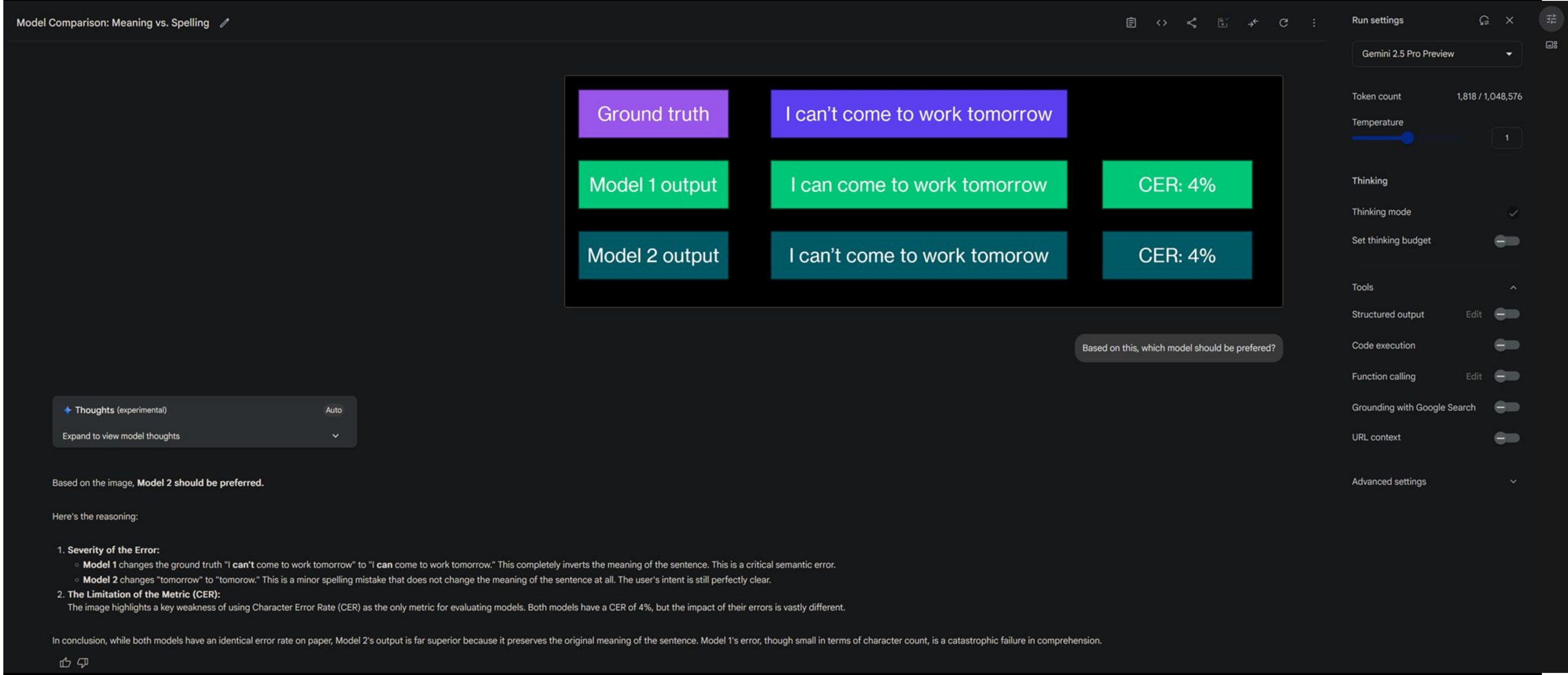



**G-EVAL: NLG Evaluation using GPT-4 with Better Human Alignment**

**Yang Liu Dan Iter Yichong Xu**
**Shuohang Wang Ruochen Xu Chenguang Zhu**

Microsoft Cognitive Services Research
{***yaliu10***, *iterdan, yicxu, shuowa, ruox, chezhu*}*@microsoft.com*

# Audio transcription

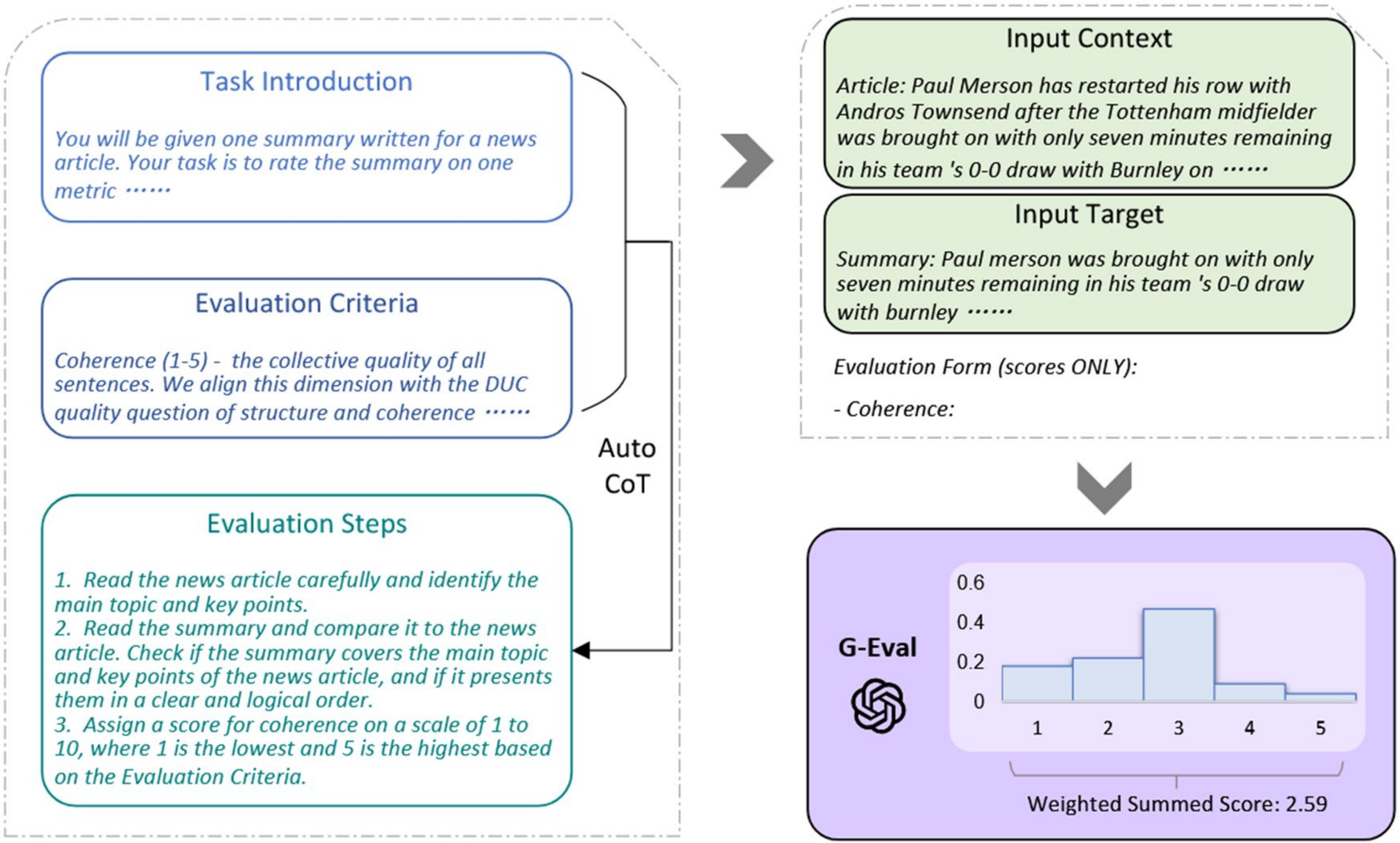


Automatic metrics (BLEU, ROUGE) correlate poorly with human judgments for creative NLG.

Reference-based scores require costly human references.

# G-EVAL Framework

1. Prompt: Task intro + detailed evaluation criteria.
2. Auto Chain-of-Thought (CoT): GPT-4 generates step-by-step evaluation instructions.
3. Scoring: Form-filling paradigm; probability-weighted aggregation yields continuous scores.


Task Introduction
You will be given one summary written for a news article. Your task is to rate the summary on one metric ······
Evaluation Criteria
Coherence (1-5) - the collective quality of all sentences. We align this dimension with the DUC quality question of structure and coherence ······
Auto CoT
Evaluation Steps
1. Read the news article carefully and identify the main topic and key points.
2. Read the summary and compare it to the news article. Check if the summary covers the main topic and key points of the news article, and if it presents them in a clear and logical order.
3. Assign a score for coherence on a scale of 1 to 10, where 1 is the lowest and 5 is the highest based on the Evaluation Criteria.
Input Context
Article: Paul Merson has restarted his row with Andros Townsend after the Tottenham midfielder was brought on with only seven minutes remaining in his team 's 0-0 draw with Burnley on ······
Input Target
Summary: Paul merson was brought on with only seven minutes remaining in his team 's 0-0 draw with burnley ······
Evaluation Form (scores ONLY):
- Coherence:
G-Eval
0.6
0.4
0.2
0
1
2
3
4
5
Weighted Summed Score: 2.59

Evaluation tasks: Text summarization & dialogue response generation.

Datasets: SummEval, Topical-Chat, QAGS.

| Metrics | Coherence | | Consistency | | Fluency | | Relevance | | AVG | |
|---|---|---|---|---|---|---|---|---|---|---|
| | $\rho$ | $\tau$ | $\rho$ | $\tau$ | $\rho$ | $\tau$ | $\rho$ | $\tau$ | $\rho$ | $\tau$ |
| ROUGE-1 | 0.167 | 0.126 | 0.160 | 0.130 | 0.115 | 0.094 | 0.326 | 0.252 | 0.192 | 0.150 |
| ROUGE-2 | 0.184 | 0.139 | 0.187 | 0.155 | 0.159 | 0.128 | 0.290 | 0.219 | 0.205 | 0.161 |
| ROUGE-L | 0.128 | 0.099 | 0.115 | 0.092 | 0.105 | 0.084 | 0.311 | 0.237 | 0.165 | 0.128 |
| BERTScore | 0.284 | 0.211 | 0.110 | 0.090 | 0.193 | 0.158 | 0.312 | 0.243 | 0.225 | 0.175 |
| MOVERSscore | 0.159 | 0.118 | 0.157 | 0.127 | 0.129 | 0.105 | 0.318 | 0.244 | 0.191 | 0.148 |
| BARTScore | 0.448 | 0.342 | 0.382 | 0.315 | 0.356 | 0.292 | 0.356 | 0.273 | 0.385 | 0.305 |
| UniEval | 0.575 | 0.442 | 0.446 | 0.371 | 0.449 | 0.371 | 0.426 | 0.325 | 0.474 | 0.377 |
| GPTScore | 0.434 | – | 0.449 | – | 0.403 | – | 0.381 | – | 0.417 | – |
| G-EVAL-3.5 | 0.440 | 0.335 | 0.386 | 0.318 | 0.424 | 0.347 | 0.385 | 0.293 | 0.401 | 0.320 |
| - Probs | 0.359 | *0.313* | 0.361 | *0.344* | 0.339 | *0.323* | 0.327 | *0.288* | 0.346 | *0.317* |
| G-EVAL-4 | **0.582** | **0.457** | **0.507** | **0.425** | **0.455** | **0.378** | **0.547** | **0.433** | **0.514** | **0.418** |
| - Probs | 0.560 | *0.472* | 0.501 | *0.459* | 0.438 | *0.408* | 0.511 | *0.444* | 0.502 | *0.446* |
| - CoT | 0.564 | 0.454 | 0.493 | 0.413 | 0.403 | 0.334 | 0.538 | 0.427 | 0.500 | 0.407 |

Table 1: Summary-level Spearman ($\rho$) and Kendall-Tau ($\tau$) correlations of different metrics on SummEval benchmark. G-EVAL without probabilities (*italicized*) should not be considered as a fair comparison to other metrics on $\tau$, as it leads to many ties in the scores. This results in a higher Kendall-Tau correlation, but it does not fairly reflect the true evaluation ability. More details are in Section 4.

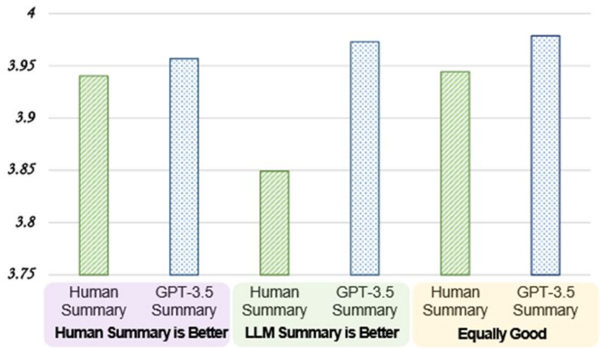



Figure 2: Averaged G-EVAL-4's scores for human-written summaries and GPT-3.5 summaries, divided by human judges' preference.

1. NLG outputs from high-quality systems are in natural difficult to evaluate. The authors of the original paper found that inter-annotator agreement on judging human-written and LLM-generated summaries is very low, with Krippendorff's alpha at 0.07.

2. G-EVAL may have a bias towards the LLM-generated summaries because the model could share the same concept of evaluation criteria during generation and evaluation.

# Ivona Najdenkoska

## DRCT: Diffusion Reconstruction Contrastive Training towards Universal Detection of Diffusion Generated Images

Baoying Chen [*1] Jishen Zeng [*1] Jianquan Yang [2] Rui Yang [1]

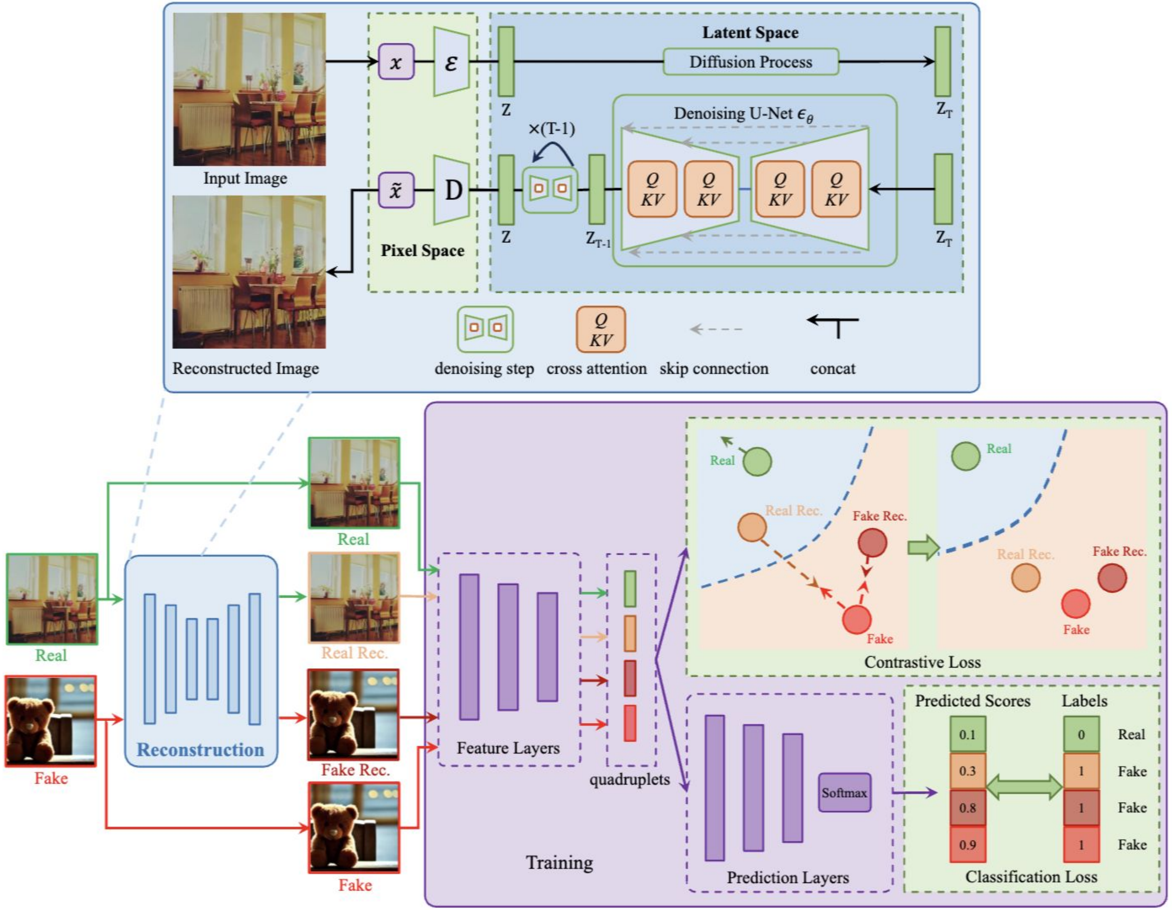
ICML 2024

# Intro

- The diffusion reconstruction on real images can preserve the image content while leaving the ***fingerprint*** of the diffusion model on the output images.
- These reconstructed images can serve as ***informative yet hard samples*** for detectors to learn the subtle differences between real and generated images.

- This paper proposes a novel training framework named **Diffusion Reconstruction Contrastive Training (DRCT).**
- DRCT significantly improves the detection accuracy and generalization ability of diffusion-generated image detectors.

# Diffusion Reconstruction Contrastive Training (DRCT)




DRCT consists of a reconstruction stage and a training stage:

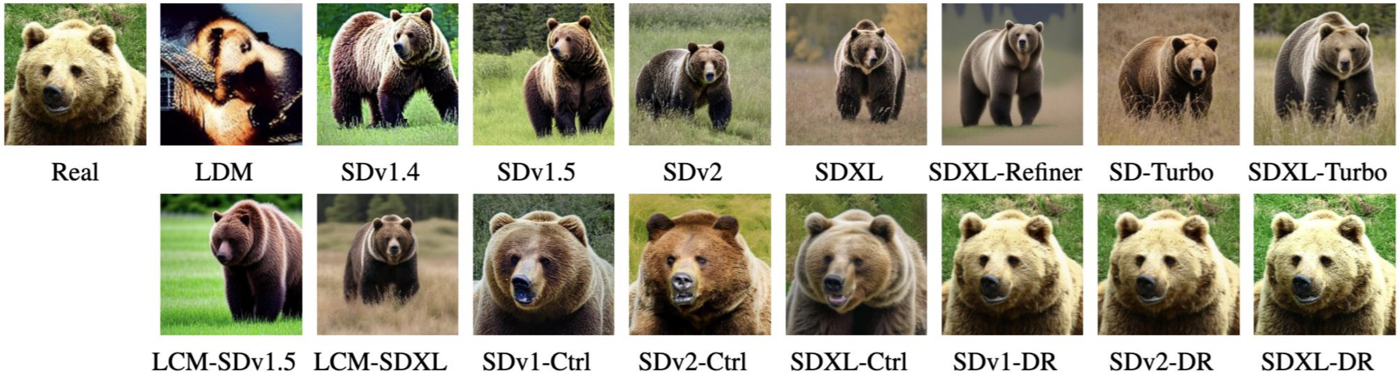
1. **Reconstruction stage:** a large number of images are produced by reconstructing both real and generated image using selected diffusion models, which are then prepared for the training of the classifier.
2. **Training stage**: 4 types of samples: real images, real reconstructed images, fake images, and fake reconstructed images, are used for computing the contrastive loss and the classification loss.

# DRCT-2M Dataset

- Collection of 2 million images for training and evaluation. It consists of two parts:
    - Images automatically generated by diffusion-based models (prompts are derived from the MSCOCO)
    - Images collected from real-world scenarios (Midjourney and CIVITAI)

The **DRCT-2M** dataset involves ***16 types of stable diffusion models***, including LDM, SDv1.4, SDv1.5, SDv2, SDXL, SDXL-refiner, SD-Turbo, SDXL-Turbo, LCM-SDv1.5, LCM-SDXL, SDv1-Ctrl, SDv2-Ctrl, SDXL-Ctrl, SDv1-DR, SDv2-DR and SDXL-DR.

The prompt used for image generation is "*A big burly grizzly bear is shown with grass in the background.*"



# Some experimental details

**Data**: The compared methods are trained on the DRCT-2M dataset (utilizing real images from MSCOCO) and the GenImage.

**Evaluation metric**: Accuracy (ACC) as the metric to evaluate detection performance, using a threshold of 0.5.

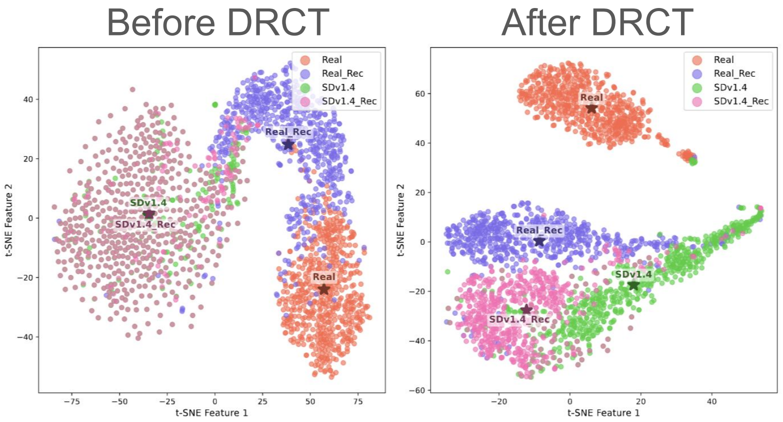



*Table 1.* Accuracy (ACC, %) comparisons of our DRCT and other generated image detectors on DRCT-2M. Except for DIRE and DRCT, all methods are only trained on SDv1.4 and then evaluated on different testing subsets on DRCT-2M. For the training data of DIRE and DRCT, when the Diffusion Reconstructed (DR) model is SDv1, the original fake images were generated by SDv1.4. When the DR model is SDv2, the original fake images were generated by SDv2.

| Method | DR | SD Variants | | | | | | Turbo Variants | | LCM Variants | | ControlNet Variants | | | DR Variants | | | Avg. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | LDM | SDv1.4 | SDv1.5 | SDv2 | SDXL | SDXL-Refiner | SD-Turbo | SDXL-Turbo | LCM-SDv1.5 | LCM-SDXL | SDv1-Ctrl | SDv2-Ctrl | SDXL-Ctrl | SDv1-DR | SDv2-DR | SDXL-DR | |
| CNNSpot | - | 99.87 | 99.91 | 99.90 | 97.55 | 66.25 | 86.55 | 86.15 | 72.42 | 98.26 | 61.72 | 97.96 | 85.89 | 82.84 | 60.93 | 51.41 | 50.28 | 81.12 |
| F3Net | - | 99.85 | 99.78 | 99.79 | 88.66 | 55.85 | 87.37 | 68.29 | 63.66 | 97.39 | 54.98 | 97.98 | 72.39 | 81.99 | 65.42 | 50.39 | 50.27 | 77.13 |
| CLIP/RN50 | - | 99.00 | 99.99 | 99.96 | 94.61 | 62.08 | 91.43 | 83.57 | 64.40 | 98.97 | 57.43 | 99.74 | 80.69 | 82.03 | 65.83 | 50.67 | 50.47 | 80.05 |
| GramNet | - | 99.40 | 99.01 | 98.84 | 95.30 | 62.63 | 80.68 | 71.19 | 69.32 | 93.05 | 57.02 | 89.97 | 75.55 | 82.68 | 51.23 | 50.01 | 50.08 | 76.62 |
| De-fake | - | 92.1 | 99.53 | 99.51 | 89.65 | 64.02 | 69.24 | 92.00 | 93.93 | 99.13 | 70.89 | 58.98 | 62.34 | 66.66 | 50.12 | 50.16 | 50.00 | 75.52 |
| Conv-B | - | **99.97** | **100.0** | **99.97** | 95.84 | 64.44 | 82.00 | 80.82 | 60.75 | 99.27 | 62.33 | 99.80 | 83.40 | 73.28 | 61.65 | 51.79 | 50.41 | 79.11 |
| UnivFD | - | 98.30 | 96.22 | 96.33 | 93.83 | 91.01 | 93.91 | 86.38 | 85.92 | 90.44 | 88.99 | 90.41 | 81.06 | 89.06 | 51.96 | 51.03 | 50.46 | 83.46 |
| DIRE | SDv1 | 98.19 | 99.94 | 99.96 | 68.16 | 53.84 | 71.93 | 58.87 | 54.35 | **99.78** | 59.73 | 99.65 | 64.20 | 59.13 | 51.99 | 50.04 | 49.97 | 71.23 |
| DIRE | SDv2 | 54.62 | 75.89 | 76.04 | **99.87** | 59.90 | 93.08 | **99.77** | 57.55 | 87.29 | 72.53 | 67.85 | 99.69 | 64.40 | 49.96 | 52.48 | 49.92 | 72.55 |
| DRCT/Conv-B (ours) | SDv1 | 99.91 | 99.90 | 99.90 | 96.32 | 83.87 | 85.63 | 91.88 | 70.04 | 99.66 | 78.76 | **99.90** | 95.01 | 81.21 | **99.90** | 95.40 | 75.39 | 90.79 |
| DRCT/Conv-B (ours) | SDv2 | 99.66 | 98.56 | 98.48 | 99.85 | 96.10 | **98.68** | 99.59 | 83.30 | 98.45 | 93.78 | 96.68 | **99.85** | **97.66** | 93.91 | **99.87** | **90.39** | **96.55** |
| DRCT/UnivFD (ours) | SDv1 | 96.74 | 96.26 | 96.33 | 94.89 | **96.24** | 93.46 | 93.43 | 92.94 | 91.17 | 95.01 | 95.60 | 92.68 | 91.95 | 94.10 | 69.55 | 57.43 | 90.49 |
| DRCT/UnivFD (ours) | SDv2 | 94.45 | 94.35 | 94.24 | 95.05 | 95.61 | 95.38 | 94.81 | **94.48** | 91.66 | **95.54** | 93.86 | 93.48 | 93.54 | 84.34 | 83.20 | 67.61 | 91.35 |

# Takeaways

- The paper proposes a universal framework - **Diffusion Reconstruction Contrastive Training (DRCT)**, for enhancing the generalizability of existing methods for detecting diffusion-based generated images.

- While DRCT also boosts the detection accuracy for GAN-generated images, the improvement is not as ***significant***.
- This difference is mainly due to the ***unique generative artifacts*** produced by GANs versus those produced by diffusion-based methods - opportunity for future work :)