

## Review 151, Short: FreeU: Free Lunch in Diffusion U-Net, 22.09.2023

<https://huggingface.co/papers/2309.11497>

אם אתם אוהבים מודלי דיפוזיה אтем תאהבו את המאמר הזה המציע شيئا' קטן די פשוט למודלי דיפוזיה גנרטיביים שմביאו אליו שיפור ניכר באיכות התמונות המוגנרטות. אז #shorthebrewpaperreviews מאמר שחקיר מה קורה בתוך המודל לשערוך הרעש המהווה לב של מודלי דיפוזיה גנרטיביים.

אזכיר כי מודל דיפוזיה מאומן לגנרט נתונים טהור על ידי הסרה מנות קטנות של רעש כל פעם (אייטרציה). הקולט למודל זהה הוא הפלט של האיטרציה וקודמת ומספר האיטרציה (שעוברת קידוד לפניו). המודל שוחזה את הרעש הוא כМОון רשת ניורונים בסגנון UNet עם כל מיני שכליולים קלים (כמו הוספה attention). UNet לוקח את הייצוג הלטנטי של הדאטה, דוחס אותו עוד יותר ואז מחזיר את הדאטה לאודל המקורי.

וכמוון יש שם את הדאטה skip-connection שמעתיק את הדאטה משלב הקטנת המיםד לשלב הגדלת המיםד של UNet. אז המחברים שמו לב שהדאטה שמגייע מה skip connection אחראי על התדרים הגבוהים בתמונה כלומר על הפרטים הקטנים של התמונה כאשר החלק השני (backbone) אחראי על פרטים משמעוניים יותר של התמונה. המחברים מצאו כי הגברת רכיב backbone לצדי החלשת רכיב הoption skip-connection מוביל לשיפור איכות התמונה המוגנרטת.

אבל מתברר שכאל הగברת מתבצע בכל הערכאים (channels) אז איקות התמונה נפגעת המגבירים רק על חצי (אין לי מושג למה) של הערכאים. לגבי התוכן של skip-connection ההחלשה מתבצעת לתדרים הגבוהים בו. כלומר עושים התמורה פוריה מחלשים את התדרים הגבוהים ועושים התמורה פוריה הפוכה. זה כל הרעיון: פשוט ואלגנטiy שנייתן למשמעותו בכמה שורות קוד בודדות.

## Review 152: CoCA: Fusing Position Embedding with Collinear Constrained Attention in Transformers for Long Context Window Extending, 23.09.2023

<https://arxiv.org/abs/2309.08646.pdf>

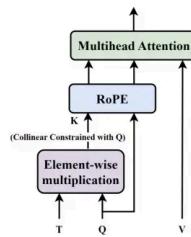


Figure 1: Collinear Constrained Attention.  $Q$ ,  $T$  and  $V$  are generated by projection matrices the same as before, we generate  $K$  with  $Q$  and  $T$  to applying collinear Constraint.

זה לא סוד שאחד המרכיבים החשובים בטרנספורמרים הינו קידוד תלוי המיקום (positional encoding) או PE. PE הוא לקודד מיקום הטוקנים בסדרה והשיטות קידוד שהפכה להיות מאוד פופולרית לאחרונה נקראת RoPE (rotary PE).

از היום ב-shorthereviews.thebrewpaper.org אנו סוקרים מאמר שמצא מקום בו ניתן לשפר (לפי המאמר) את RoPE ומציע דרך לפטור אותה. אז קודם כל מה זה RoPE? זו שיטה שלמעה לוקחת וקטורי שאלתה ומפתח (query and key) וככפילה אותם (איבר איבר) בזוקטור מרוכב בעל נורמה יחידה שהתדר שלו פרופורציוני למיקום של טוקן בסדרה (כל איבר בזוקטור זה מוכפל גם במייד שלו במרחב הייצוג).

כלומר ככל שהטוקן נמצא יותר רחוק מתחילה הסדרה התדר שלו (מקדם מערכי במספר המרכיב זהה) הינו גבוה יותר. ציריך לציין שוקטורי המפתח והשאלתה מיוצגים כוקטורים מרוכבים גם כן. כאשר מחשבים את  $\text{attention}$ -between וקטורים אלו יוצא כי יש פונקציה  $\text{attention}$ -between תלויה באופן מפורש במרחק בין וקטורים אלו (נמצא בתוך אקספוננטה מרוכבת).

ניתן להוכיח ככל שעבור מרחק גדול בין הטוקנים  $\text{attention}$  ביניהם שואף לאפס. עכשו המחברים שמו לב שעבור ממדים מסוימים במרחב הייצוג ציוני  $\text{attention}$  בין מקדמי שאלתה ומפתח (עבור כל מימד מדובר בשני זוגות של מספרים מרוכבים) עלולים לכך כאשר מרחק בין מיקומי הטוקנים קטן (בגלל המבנה של RoPE). כמובן שהוא לא רצוי ולמרות זהה קורה רק לממדים מסוימת המחברים מוכחים זהה משפיע לרעה על יעילות הקידוד המיקומי.

הסיבה לכך (הנובעת מאריתמטיקה די פשוטה) היא הזריות שהוא לא אפס בין וקטורי השאלתה והמפתח. אך המחברים מציעים שיטה ההופכת וקטורי שאלתה להיות קולינאריים ככלומר הזריות ביניהם הופכת להיות 0 והבעיה נעלמת. יש שיפור מסוים בביטויים אך המחברים עצם אומרים שטרם סימנו לבדוק את כל ההיבטים של הגישה המוצעת.

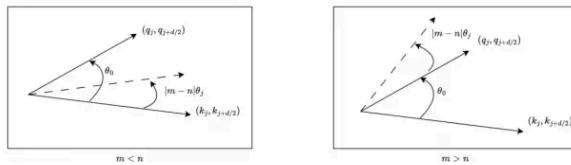
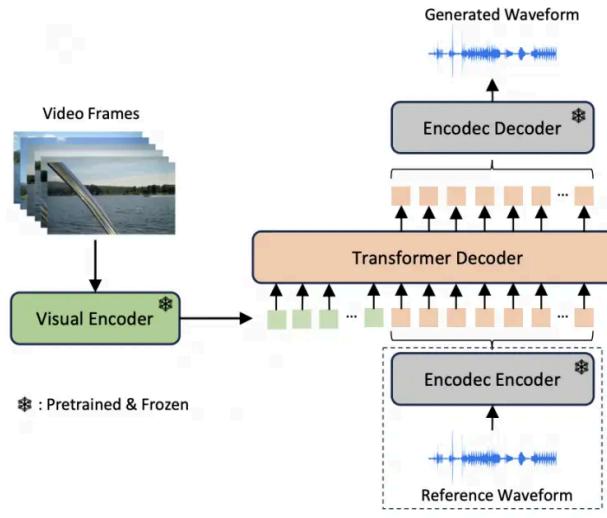


Figure 2: Broken order in bidirectional models. The inner-product of  $\mathbf{q}_j$  and  $\mathbf{k}_j$  is based on the relative angle between them. **Left:** when  $|m - n|\theta_j < \theta_0$ , the relative angle between  $\mathbf{q}_j$  and  $\mathbf{k}_j$  will decrease with  $|m - n|$ , which means closest tokens may get smaller attention scores. (We use 'may' here since the attention score is the sum of  $d/2$  inner-products, maybe one of them is insignificant. However, experiments confirmed this significance.). **Right:** no anomalous behavior.

## Review 153, Short: FOLEYGEN: VISUALLY-GUIDED AUDIO GENERATION, 24.09.2023

<https://huggingface.co/papers/2309.10537>



**Fig. 1.** Overview of the FoleyGen system. The dashed-line block shows the EnCodec encoder for converting waveforms into discrete tokens, utilized only during training.

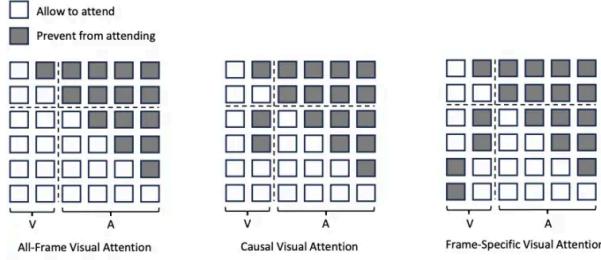
בחרתי לסקור את המאמר זהה כי למדתי ממנו שיש שימושה שלא הכרתי בראיה ממוחשבת והיא הפקה סדרת אודיו מסרטון וידאו נתון. מתברר שמדובר במקרה לא טריוויאלי והמאמר מציע גישה אלגנטית ודי פשוטה לבנייה זו.

از היום ב- #shorthereviews שוקרים מאמר המציע שיטה להפקה אודיו מווידאו. בהינתן נתונים המכיל זוגות של סרטוני וידאו או אודיו המתאים המחברים משתמש באנקודר מאומן של אודיו EnCodec שהופך את האודיו לייצוג הלטנטי. מה זה ייצוג לטנטי של אודיו?

למעשה זו סדרה של וקטורים שכל אחד מהם הוא השיקון (embedding) של מقطع (זמן) של אותן. בנוסף יש EnCodec דקודר שמשחזר את אותן מהייצוג הלטנטי שלו. המאמר גם משתמש במודלים שמטרתם להפוך ייצוג של וידאו (של כל פרוי) כמו CLIP, ImageBind ו-ViT.

از מה בעצם הארכיטקטורה של FoleyGen ואיך מאמנים את המודל הזה? לכל זוג של וידאו וידאו מעבירים את האודיו דרך האנדקור של EnCodec ואת הVIDeo דרך האנקודר של DATA ו-VIDeo (VID). ככלומר כאן אודיו וידאו מוצגים באמצעות סדרה של וקטורי הייצוג של "טוקנים" שמרכיבים אותם (פריטים לוידאו ומقطع זמן לאודיו).

לאחר מכן מאמנים טרנספורמר (モジュולר מתקודר בלבד) שמטרתו לשחזר את ייצוג הטוקן הבא של אודיו בהינתן ייצוג הטוקנים (של אודיו) הקודמים וייצוג של טוקני הוידאו. הם בוחנו כמה אופציות לגבי טוקנים של הוידאו של הטרנספורמר יכול לעשות: כל הטוקנים, רק הטוקנים שבאו לפני הזמן או את טוקני הוידאו הסוכרים בזמן. זה זה – פשוט ואלגנטי.



**Fig. 2.** Overview of the three visual attention mechanisms. For simplicity, here we assume we have 2 visual features ‘V’ and 4 audio tokens ‘A’ with a frame rate of 2 Hz.

## Review 154: Context is Environment, 26.09.2023

<https://arxiv.org/abs/2309.09888>

### סקירה זו נכתבת על ידי עדן יבין

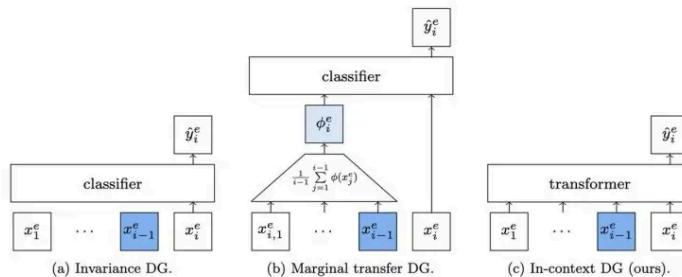


Figure 1: Three frameworks for domain generalization (DG), predicting the target  $y_i^e$  from the input  $x_i^e$  at test environment  $e$ . Depicted in blue, the last example  $x_{i-1}^e$  contains relevant features for the current prediction. (a) Invariance DG discards all of the previously observed information from the test environment, removing too much predictive signal. (b) Marginal transfer DG summarizes all of the previously observed test inputs as a coarse embedding, diluting predictive signal found at the example level. (b) Our in-context DG directly observes all of the previous test inputs, allowing the search of “needle-in-the-haystack” signals, such as the relevant one in  $x_{i-1}^e$ .

מודל של רכב אוטונומי צריך לדעת להתמודד עם המון מצבי עולם אשר לא ראה בזמן תהליכי האימון. כיצד הוא עושה זאת? מזעור הטעות על דוגמאות כאלו הינו בתחום שלם הנקרא Domain Generalization. האם מודלי שפה יוכל לעזור בתחום זה ולהראות שיפור על פני המצביעים?

נגלת היום ב-#shorthereviewpaperreviews Context is Environments. המאמר נקרא ובסגנון החוקרים מנסים להראות שמה שקוראים לו “הסבירה” בתחום ה-DG מתקבל לא-*in-context* בתחום של מודלי שפה. שיטות קיימות בתחום ה-DG מנסות להשתמש בעבר שנוצרו מאינטראקציה עם הסביבה כדי לחזות את התוצאה של הדגימה הנוכחיית.

אך האם לא כך גם מודלי שפה? הרי הם משתמשים בטוקנים הקודמים כדי לחזות את הטוקן הנוכחי. יותר מכך, עם השימוש ההולך וגובר במודלי שפה גילו את היכולת שלהם ללמידה *in-context* באמצעות טכניקות כגון few-shot. נוכל להשתמש בכך בשכיל לשפר את יכולת ההלכה של מודלים אלו על דוגמאות אשר לא רואו.

השיטה של החוקרים נקראת ICRM, ובקצרה מנסה לשתמש בكونטקט סובייה כדי להקטין את הסיכון לטעויות על דוגמאות אשר לא נראו ולא דומות למה שהיא באימון המודל.

כאשר מודל השפה  $\hat{y}$  מנסה לשערק את  $(C|X|Y)$  על ידי שימוש בפונקציית הפסד של cross-entropy loss. השערוק של  $(P|X|C|Y)$  הינו בשביל לשערק את הסיכון של טעות בחיזוי בהינתן הדוגמא הנוכחית והסביבה או הקונטקט.

החוקרים מראים שימוש פשוט זה מביא לתוצאות טובות יותר מהשיטות הקודמות בנים'ים הכללים יכולת הכללה על דוגמאות חדשות שלא נראו בסט האימון. למי שירצה להתעמק יותר, המאמר מראה עוד המון נקודות קרייטיות וחוובות בשימוש של מודלי שפה בשביל לחשב סיכון של דוגמאות חדשות ובנוסף נותן עוד תאוריה מעניינת על התחום.

section knits these two threads together, enabling us to attack the problem of domain generalization with in-context learners. The plan is as follows:

- Collect a dataset of triplets  $\mathcal{D} = \{(x_i, y_i, e_i)\}_{i=1}^n$  as described in Section 2. Initialize a next-token predictor  $\hat{y} = h(x; c)$ , tasked with predicting a target label  $y$  associated to the input  $x$ , as supported by the context  $c$ .
- During each iteration of training, select  $e \in \mathcal{E}_{\text{tr}}$  at random. Draw  $t$  examples from this environment at random, construct one input sequence  $(x_1^e, \dots, x_t^e)$  and its associated target sequence  $(y_1^e, \dots, y_t^e)$ . Update the next-token predictor to minimize the auto-regressive loss  $\sum_{j=1}^t \ell(h(x_j^e; c_j^e), y_j^e)$ , where the context is  $c_j^e = (x_1^e, \dots, x_{j-1}^e)$ , for all  $j = 2, \dots, t$ , and  $c_1^e = \emptyset$ .
- During test time, a sequence of inputs  $(x'_1, \dots, x'_{t'})$  arrives for prediction, one by one, all from the test environment  $e' \in \mathcal{E}_{\text{te}}$ . We predict  $\hat{y}'_j = h(x'_j, c'_j)$  for  $x'_j$ , where the context  $c'_j = (x'_1, \dots, x'_{j-1})$ , for all  $j = 2, \dots, t'$ , and  $c'_1 = \emptyset$ .

We call the resulting method, illustrated in Figure 1c, In-Context Risk Minimization (ICRM).

## Review 155, Short: CHAIN-OF-VERIFICATION REDUCES HALLUCINATION IN LARGE LANGUAGE MODELS, 27.09.2023

<https://arxiv.org/abs/2309.09888.pdf>

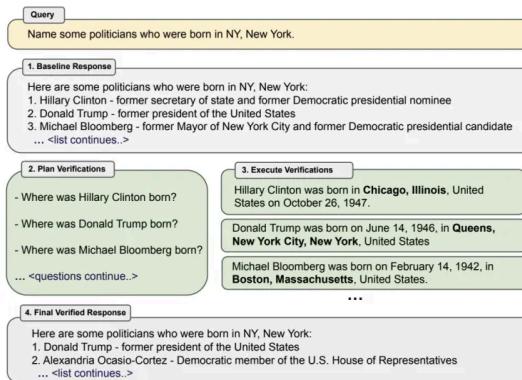


Figure 1: Chain-of-Verification (CoVe) method. Given a user query, a large language model generates a baseline response that may contain inaccuracies, e.g. factual hallucinations. We show a query here which failed for ChatGPT (see section 9 for more details). To improve this, CoVe first generates a plan of a set of verification questions to ask, and then executes that plan by answering them and hence checking for agreement. We find that individual verification questions are typically answered with higher accuracy than the original accuracy of the facts in the original longform generation. Finally, the revised response takes into account the verifications. The factored version of CoVe answers verification questions such that they cannot condition on the original response, avoiding repetition and improving performance.

מכירים את בעיית ההזיות (hallucinations) במודלי שפה? בגודל זה קורה כאשר מודל שפה מספק לנו תשובות לא נכונות לשאלות לפעמים יחסית פשוטות. סוגיה זו קיילה התייחסות רבה לאחרונה במספר עבודות ומאמר שנסקרו היום ב-#shorthereviewspaperstheshortbrewpaperreviews מציע גישה נוספת לפתרונה.

הרעיון של עליו מtabסתה השיטה המוצעת במאמר הוא פשוט ומאוד אינטואיטיבי ומסתמך על אוביツרציה הבהא: יש לא מעט מקרים שמודלי שפה מספקים לנו תשובות לא נכונות אך כאשר מבקשים מהם לבדוק את התשובה הוא חוזר בו ונוטן תשובה נכונה (אצין שלא תמיד זה קורה מסיבות טובות אלא נבע מاؤפן האימון שלו הגורם למודל להיות קצת "yesman" עם האנשים).

אבל אולי כדאי לנו לבקש מהמודול לבדוק את תשובתו על ידי גנרטו רלוונטיות לתשובה. למשל אם המודלעונה על שאלה איזה פוליטיקאים ידועים נולדו בשנות השישים, והוא עונה X אז אחת שאלות הבדיקה יכולה להיות "מתי נולד פוליטיקאי X". על סמך תשובות על השאלה אלו (שכמובן יכולת להיות רבות ומגוונות) המודל משנה את תשובתו והופך אותה למדויקת יותר. זה וזה – כל השאר זה הנדסת פרומפרטים 5-5 השלבים הבאים:

- שאלה מקורית
- תשובה הIGINITALית של מודל שפה
- גנרטו שאלות וריפקציה לשאלת הIGINITALית וגנרטו תשובות עליהם (מתבצע בשני שלבים)
- מודיפיקציה של התשובה הסופית בהתבסס על התוצאה של שלב הקודם.

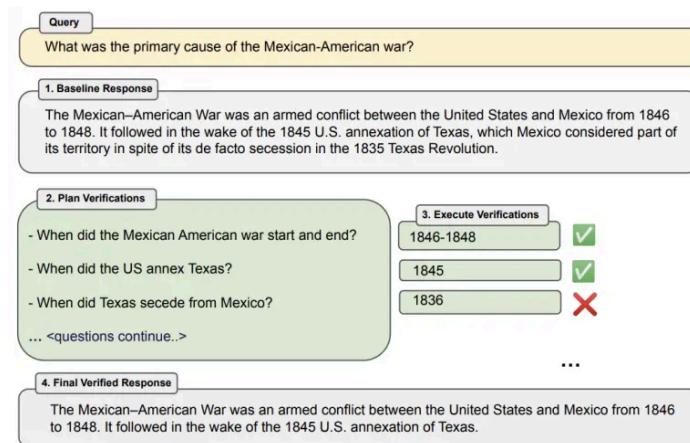


Figure 3: For longform generation, the Chain-of-Verification (CoVe) Factor + Revise method is the most effective in our longform generation experiments. CoVe Factor + Revise has the model independently identify (cross-check) which facts are consistent with its executed verifications (indicated by tickmark and crosses in the figure). With this extra step we aim to disregard the inconsistent facts and use the consistent facts to regenerate the response.

## Review 156, Short : LONGLORA: EFFICIENT FINE-TUNING OF LONG CONTEXT LARGE LANGUAGE MODELS, 28.09.2023

<https://arxiv.org/abs/2309.12307.pdf>

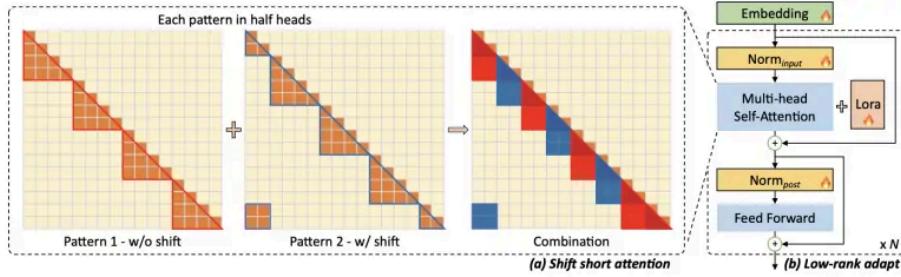


Figure 2: Overview of LongLoRA designs. LongLoRA introduces shift short attention during fine-tuning. The trained model can retain its original standard self-attention during inference. In addition to plain LoRA weights, LongLoRA additionally makes embedding and normalization layers trainable, which is essential to long context learning, but takes up only a small proportion of parameters.

כל מי שעוסק במודלי שפה בטח שמע על סוגיית אורך הקשר (context length). אנו רוצים שהמודלים שלנו יהיו מסוגלים “להחזיק בבטן” כמה שיותר גודלה של טקסט. אולם הקשר ארוך דורש כמות עצומה של משאבים לאימון ולאינפראנו.

از היום ב-#shorthebrewpaperreviews אנו סוקרים מאמר שמציע גישת טיבוב(fine-tuning) שגדילה את אורך הקשר של מודל שפה. ככלומר אם מודל שפה היה מאומן באימון מקדים(pretraining) עם אורך הקשר של 2048, השיטה המוצעת מאפשרת להאריכו פי 4 ל-8192. כמו שאתם רואים השם של השיטה מכיל את המילה LoRA שהיא שיטה מאוד פופולרית לפינ-טיוון של מודלי שפה.

במקום לכיל (לשנות) את כל המשקלים של מודל השפה המכיל LoRA מעדכנת רק את התוספת למשקלים המודול (כמו ResNet). בסוף התוספת למשקלים מיצגת על ידי מטריצה עם ראנק נמוך שנייתן לתאר אותה על ידי מכפלה של מטריצות בעלות מידת נמוך יחסית.

אז מה מציע LongLoRA בנוסף? כדי להגדיל את אורך הקשר נגד מ-2048 ל-8192 היא מחלקת את 8192 טוקנים ל-4 קבוצות בעלות 2048 טוקנים כל אחת שעבור כל אחת מהם צינוי-h-attention מחושבים בנפרד (חיסכון פי 16 בחישובים). את זה עושים בחצי מחראשים. בשאר הראשים פשוט מזידים את הקבוצות האלו בחצי גודל ככלمر הקבוצה הראשונה תכלי טוקנים מ-1024 עד 3072, השניה מ 3072 עד 5195 וכדומה. טרייך פשוט מאד אבל מביא תוצאות לא רעות בכלל.

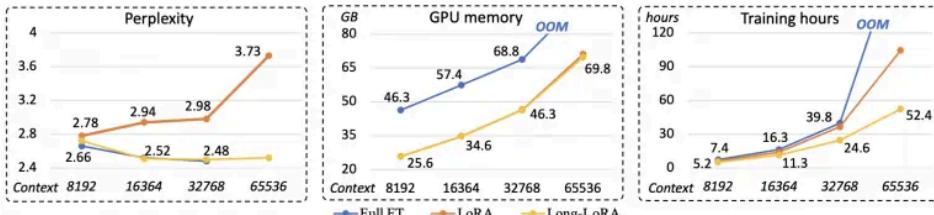
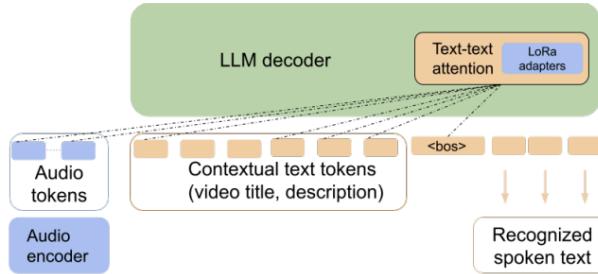


Figure 1: Performance and efficiency comparison between full fine-tuning, plain LoRA, and our LongLoRA. We fine-tune LLaMA2 7B on various context lengths, with FlashAttention-2 (Dao, 2023) and DeepSpeed (Rasley et al., 2020) stage 2. Perplexity is evaluated on the Proof-pile (Azerbayev et al., 2022) test set. Plain LoRA baseline spends limited GPU memory cost, but its perplexity gets worse as the context length increases. LongLoRA achieves comparable performance to full fine-tuning while the computational cost is much less.

# Review 157, Short: End-to-End Speech Recognition Contextualization with Large Language Models, 30.09.23

<https://huggingface.co/papers/2309.10917>



**Fig. 1.** A speech recognition model with mixed-modal context consisting of audio and optional text tokens based on a pretrained LLM backbone. Speech encoder and LLM decoder are both initially pretrained. The LLM weights are frozen (orange blocks), while audio encoder and LoRa adapters are fine-tuned during training (blue blocks).

אודיו וטקסט? נכוּ ששני סוגים דатаה אלו הם די שונים ולא הגיוני להניח שמודל שאומן על טקסט יכול להיביא תוצאות טובות גם על אודיו לאחר כיוול קל. אך התבර שזה אפשרי. במאמר שנסקרו היום ב-#shortthebrewpapereviews#. מעשה לךו מודל שפה מאומן והשתמשו בו בשבייל לבצע משימה audio2text. כלומר להפיק את מה שנאמר בקטע אודיו.

air המ עשו זאת? מכיווןuai אפשר סתם לחת את אודיו להזין אותו כמו שהוא למודל שפה גדרש כאן אנקדור שמקודד את הפיצרים המהותיים של אות אודיו. מחברי המאמר משתמשים במודל מאומן מראש הנקרא ConFormer ומפיק לנו ייצוג לטנסי של אות אודיו (כלומר מערך של וקטורים המייצגים כל מقطع של אודיו או בפשטות טוקני אודיו). ד"א ConFormer הוא מודלIDI מיעניין (הוציא על ידי גוגול) המשלב ארכיטקטורת הטרנספורמר עם שכבות קונבולוציה (משתמשים שם גם בקידוד מיקום יחס RoPE שניהה מאוד פופולרי היום).

לאחר מכן לוקחים את ייצוג של טוקני האודיו ומזינים אותו למודל שפה מאומן (המלקחו LLAMA) יחד עם עוד מידע על האודיו כמו שם היידאו שממנו הוא נלקח או התיאור הטקסטואלי. בסוף מט"ב(im) (fine-tune) מודל שפה בסגנון LoRA על דאטסהט המורכב מזוגות של אודיו והtekst. ולהפתעתך זה עובד ממש לא רע.

**Table 2.** WER under different context perturbations during decoding stage.

Context noise	WER (%)	Rare WER (%)
(Original context)	11.22	23.88
(Remove all context)	11.98	28.64
Random	12.07	28.85
Respellings	11.89	28.31
Respellings (append)	11.46	25.59
Ground Truth	10.50	19.54

# Review 158, Short: Linguistic Binding in Diffusion Models: Enhancing Attribute Correspondence through Attention Map Alignment

<https://arxiv.org/abs/2306.08877>



Figure 1: Visual bindings of objects and their attributes may fail to match the linguistic bindings between entities and their modifiers. Our approach, SynGen, corrects these errors by matching the cross-attention maps of entities and their modifiers.

מודל דיפוזיה מודרני מצטיין ביצירת תמונות באיכות מרתקת טקסטואלי (ובטח DALLe3) וברב המקרים התמונה ממש מתאימה לתיאור. אולם עדין יש מקרים שמודל מתבלבל למשל בין הצבעים של האובייקטים המופיעים בתיאור. היום ב-#shortthebrewpaperreviews המציע שיטה למניעת הבלבול סמנטי בין תכונות האובייקטים בתמונה.

הגישה המוצעת הינה פשוטה ואלגנטית. בשלב הראשון המחברים בונים את גרפ התלות הסינטקטית של הפרופט כלומר מפיקים את כל קבוצות המילים (נגיד שם עצם ושם תואר) המתאיםים אחד לשני (כמו (ארנב, צהוב) או (קורסא, בסגנון, מלון)). לאחר מכן המחברים מכילים מודל שפה עם פונקציה לוס ש"מפקחת" על הדיקןsemantic של האובייקטים בתמונה.

AIR זה נעשה? אתם בטח ידעים شيיצוג של כל מילה בפורומפט (מופק באמצעות מודל שפה) מזון למודל דיפוזיה. אך פונקציית לוס זו מנסה לאכוף דמיון בין מיללים מסוימת קבוצת שייכות (שנבנה בשלב הקודם). מפות attention אלו הם למעשה ציון cross-attention בין ייצוגי המילים (טוקנים) לבין פאטרצ'ים בתמונה.

איבר נוספת בפונקציית הלוס מכיל איבר המקיים מרחק בין מפות attention בין המילים שלא שייכים אותה קבוצה. הדמיון בין מפות attention מחושב באמצעות מרחק KL סימטרי (נקרא גם JSD). המחברים טוענים כי הלוס זה מופעל בחלק מהאיטרציות של מודל דיפוזיה - על החצי הראשון של האיטרציות.

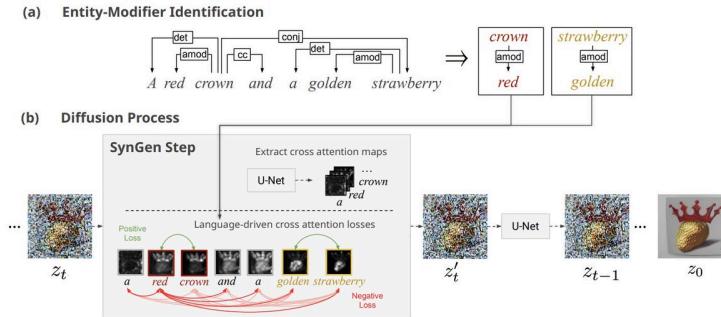


Figure 2: The SynGen workflow and architecture. (a) The text prompt is analyzed to extract entity-nouns and their modifiers. (b) SynGen adds intermediates steps to the diffusion denoising process. In that step, we update the latent representation to minimize a loss over the cross attention maps of entity-nouns and their modifiers (Eq 3).

## Review 159, Short: Stochastic Re-weighted Gradient Descent via Distributionally Robust Optimization, 02.10.2023

<https://openreview.net/forum?id=EhK6wBBJNS>

---

### Algorithm 1 Re-weighted Gradient Descent (RGD)

---

```

1: Input: Data  $\{X_i, Y_i\}_{i=1}^n$ , learning rate  $\eta$ , number of iterations  $T$ , loss function  $\ell$ , re-weighting function  $g$ , mini-batch size  $B$ 
2: for  $t = 1 \dots T$  do
3:   Sample minibatch  $\{X_i, Y_i\}_{i=1}^B$ 
4:   Compute losses for points in the minibatch:  $\ell_i \leftarrow \ell(X_i, Y_i; \theta)$ ,  $\forall i \in 1 \dots B$ 
5:   Compute per-sample weights using our proposed approach:  $w_i \leftarrow g(\ell_i) \forall i \in 1 \dots B$ 
6:   Compute the weighted pseudo-gradient:  $v \leftarrow \frac{1}{B} \sum_{i=1}^B w_i \nabla_\theta \ell_i$ 
7:   Update weights of the neural network:  $\theta \leftarrow \theta - \eta v$ 
8: end for

```

---

האם כל הדוגמאות בדאטסהט שלנו שוות? כשאתם מאנים מודל שלכם (נגיד רשת נוירונים) אתם עושים זאת עם באז'ים כאשר כל דוגמא בבאז' תורמת לעדכון המשקלים באויה מידה (לפי ערך הגרדיינט בה). אבל האם זה אופטימלי? היום ב-#shorthereviews פורסמו סקרים מאמר שמציע שיטה למשקל של תרומות הדוגמאות לעדכון משקל המודל פרופורציונלית לאקספוננט של ערך של פונקציית בה.

כלומר ככל שדוגמא קשה יותר היא תתרום יותר לשינוי משקל המודל. זה נשמע די הגיוני - כבר רأינו את הגישה הזו ב-AdaBoost לפני שנים. אבל איך המחברים הגיעו לכך? אוקי, אז קודם כל אנו מעוניינים לאמן מודל המציג את השגיאה על הדאטה שלנו.

אבל אין לנו את כל הדאטה אלא רק דאטסהט אימון. אחת הדריכים להתחשב בכך יש אין לנו רק מדגם ולא כל הדאטה היא אכןו את המודל על שונות של המודל (על הדאטה - זה קשור ל-distributional bias variance tradeoff). מכיוון שיחסוב שונות המודל הוא מורכב מאוד (ושערוכו רועש למדוי) אז משתמשים בסוג של קירוב שմזעור את הערך המקסימלי של פונקציית לוס על כל התפליגיות הדאטה הקורבות להתפלגות של דאטסהט האימון.

הקרבה מחושבת באמצעות f-divergencesKL, total variation distance או לא מעט אחרים). גישה של נקראתrobust optimization (נקראת DRO - distributional robust optimization). המחברים מצאו קשר בין DRO לבין משקל תרומה של דוגמאות בעדכון של הגרדיינט. מאמר מאד מעניין - ממליץ לצלול לעומק לפרטים המתמטיים.

**Proposition 3.1.** Consider DRO with KL-divergence-based uncertainty set. Then  $\min_{\theta \in \Theta} \widehat{R}_{D,n}$  can be rewritten as

$$\min_{\theta \in \Theta} \frac{1}{\gamma} \log \mathbb{E}_{\widehat{P}_{\text{data}}} [e^{\gamma \ell(z; \theta)}],$$

for some constant  $\gamma > 0$  that is independent of  $\theta$ .

## Review 160, Short: Vision Transformers Need Registers

<https://arxiv.org/abs/2309.16588>

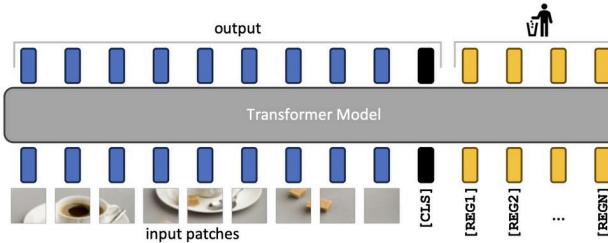


Figure 6: Illustration of the proposed remediation and resulting model. We add  $N$  additional learnable input tokens (depicted in yellow), that the model can use as *registers*. At the output of the model, only the patch tokens and CLS tokens are used, both during training and inference.

משמעותי את המאמר זה אבל לא בಗל שהוא הציע איזה רעיון מהפכני (הרעיון ד' נחמד אבל לא איזו פצחה). הסיבה לכך היא שהמאמר הזה הדגיש שוב את העבודה כמה מעט אנחנו מבינים מודל ענק בעל בילוני פרמטרים

היום ב-#shorthebrewpaperreviews סוקרים מאמר שבמילים ממש פשוטות מצא שייצוג הדadata שהטרנספורמרים הוויזואליים (כמו DINOv2) מפיקים מכילים DATA מיותרת שלא תורם לביצוע המודל יותר מדי. ראוי שהמטרה העיקרית של מודל הענק האלו היא לבנות ייצוג של DATA המכיל את הפיצרים המהותיים ביותר שלו.

כלומר הטרנספורמרים הוויזואליים לא מצליחים לקודד את המידע בצורה המיטבית יש חלקים מיוחדים בייצוג זהה. איך המחברים בכלל הגיעו להז ? הם שמו לב שיש פיצאים בתמונה שנורמה של ייצוגם (מהשכבה האחורונה) היא גודלה באופן anomalיה יחסית לייצוג הפיצאים האחרים.

המחברים גם שמו לב שייצוגים של פיצאים חריגים אלו מאוד דומים לייצוג הפיצאים הסטנדרטיים (imbalance מרחוק קווין). בנוסף יכולת של ייצוג פיצאים anomalים אלו להציג את מיקום הפיצן בתמונה היא שימושית יותר נוכח מהפיצאים הרגילים (אימנו מודל ליזיה המיקום). הם עשו עוד בדיקות נוספות ששכנעו אותם שייצוג הפיצאים אלו לא משפר את איצות המודל.

אז מה הם עושים? משחו ד' אלגנטוי (זה לא רעיון חדש כי כבר עשו זאת לפני כמה שנים במאמר על מודל שפה). אז הם הוסיפו כמה טוקנים (אחרי טוקן [cls]) שמרתם היא להכיל מידע לא רלוונטי. ייצוג טוקנים אלה פשוט נזרקיים ולא משמשים לא לאיומון ולא לאינפרנס. וזה אכן משפר את ביצועי המודל בכמה משימות.

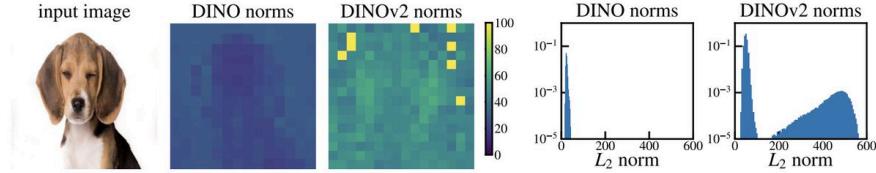


Figure 3: Comparison of local feature norms for DINO ViT-B/16 and DINoV2 ViT-g/14. We observe that DINoV2 has a few outlier patches, whereas DINO does not present these artifacts. For DINoV2, although most patch tokens have a norm between 0 and 100, a small proportion of tokens have a very high norm. We measure the proportion of tokens with norm larger than 150 at 2.37%.

## Review 161, Short: PixArt- $\alpha$ : Fast Training of Diffusion Transformer for Photorealistic Text-to-Image Synthesis

<https://arxiv.org/abs/2310.00426>

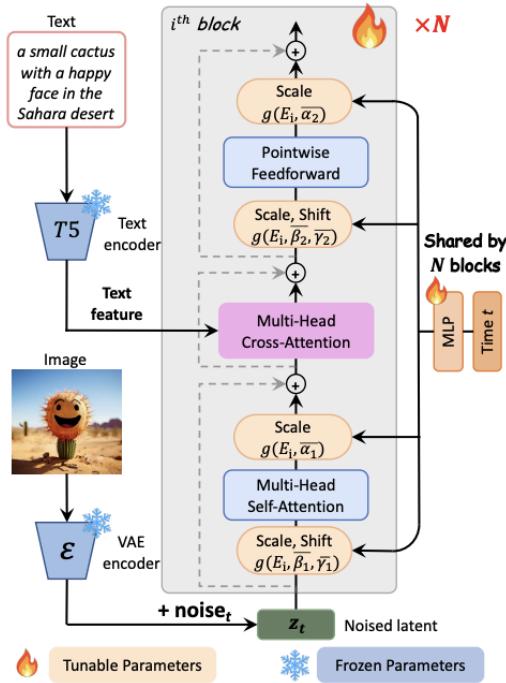


Figure 4: Model architecture of PIXART- $\alpha$ . A cross-attention module is integrated into each block to inject textual conditions. To optimize efficiency, all blocks share the same adaLN-single parameters for time conditions.

מאמר חמוד שמציע זירוז תהילכי האימון של מודלי דיפודיה שידועים כמו אוד יקרים ויוצרים כמותות גדולות של פליטת CO<sub>2</sub>. בגדול המאמר מציע לשלב כמה גישות לאימון שהוצעו בזמן האחרון. אז היום ב-#shortthebrewpaperreviews סוקרים מאמר עם השם שמכיל מילה "אומנות" (משר את תשומת ליבי ללא ספק והיווה אחת הסיבות לכך שעיני נתפסה על המאמר זהה).

אוקי', אז מה הם בעצם עושים? קודם כל הם לוקחים דאטאסת של תמונות (כמו LAION) ויצרו כותרות של התמונות בו באמצעות מודל חזק הנקרא LLaVA הטעה במאמר שכך נוצרות כותרות עשירות (סמנטיות) הרבה יותר מהדאטאסט המקורי דבר שני, הם השתמשו בארכיטקטורת הטרנספורמרים כמודל לשערוך הרעש במודל דיפוזיה (במקום UNet).

כלומר הם לוקחים מודל דיפוזיה שפועל בתחום לטנטי (stable diffusion) והחליפו UNet בהרבה שכבות של טרנספורמרים (זה הוצע במאמר Scalable Transformers למייטב ידיעתי) הם גם שכללו את שכבת הנורמל בטרנספורמר שאפשר להם לחזור את כמות הפרמטרים בצורה משמעותית. בשלב האחרון הם יכולים באמצעות high-quality aesthetic data (הם הסבירו במאמר איך הם יוצרים אותו)

שילוב של כל הגישות הללו אפשר להקטין את זמן האימון (ופליות הגז) בצורה משמעותית ד"א הם השתמשו בגרסאות המוקפות של מודל D5 כדי לבנות ייצוג של טקסט וב- VAE מהמאמר על מודלי דיפוזיה לטנטים לבניית ייצוג התמונה.



Figure 1: Samples produced by PIXART- $\alpha$  exhibit exceptional quality, characterized by a remarkable level of fidelity and precision in adhering to the provided textual descriptions.

## Review 162, Short: Think before you speak: Training Language Models With Pause Tokens

<https://arxiv.org/abs/2310.02226>

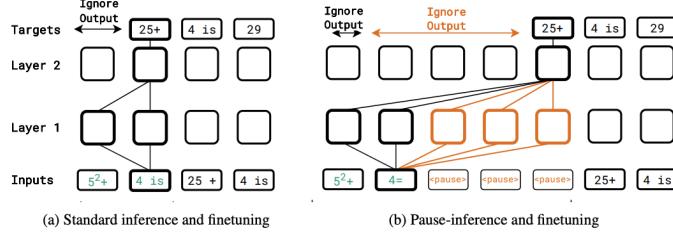


Figure 1: **Standard vs. pause-inference (and finetuning).** We consider a downstream task where, given a prefix, the decoder-only model (bidirectionally) attends to all of the prefix to generate its target answer. The rounded squares denote one Transformer operation (a self-attention and MLP) in a 2-layer Transformer. Any **Ignore Output** denotes that during inference, the corresponding output token is not extracted and thus, not fed back autoregressively; during finetuning, this output is not backpropagated through. The connecting lines denote some (not all) of the “computational pathways” within the model. Specifically, we visualize only those pathways that begin at a specific token in the prefix (here arbitrarily chosen to be “4 is”) and end at an output token (here arbitrarily chosen to be “25+”). All differences between the two settings are highlighted in color. (a) In standard inference (finetuning), the model’s output is extracted immediately upon seeing the last prefix token. (b) In pause-inference (and pause-finetuning), this is initiated only after appending a manually specified number of <pause> tokens. This introduces new computational pathways (the colored lines) between the prefix token and the output token of interest.

אחרי שלפנינו יומיים סקרנו מאמר שהכנים טוקנים שלאחר מכון “נזרקים לפח” באימון וגם באינפראנס בטרנספורמרים ויזואליים והיו המגע הזמן לסקור מאמר שהמציע טוקני “הפסיקת-pause” (גם נזרקים לפח) ויש להם מטרה קצר שונה. אז היום ב-#shortherebrewpaperreviews סקרים מאמר שמציע להשתיל טוקני הפסיקת pause המאפשרים לתת למודל שפה “הפסיקות לסידור החשיבה”.

אם זה נשמע לכם קצת משעשע אז אני איתכם אבל עובדתית הטריק המכחיק הזה מוביל לשיפור ביצועי מודלים במספר שימושות. אז איך זה עובד בעצם? זה עובד לפי סוג האימון. באימון מקדים (pretraining) משתמשים את טוקני הפסיקת pause במקומם אקראים והם לא משתתפים בחיזוי (ההסתברות המותנית של מיקטע טקסט לא תליה בהם למרות שהם בפנים).

המטרה כאן היא לאמן את הייצוג (embedding) של הטוקנים האלו. בטuib (fine-tuning) ובאינפראנס מכניםים את טוקני הפסיקת pause האלו אחריו הפורמופט במטרה לתת למודל “סוג של קצר זמן לחישוב ולסן מידע לא רלוונטי”: כמובן שאין חיזוי עבור טוקנים אלו גם כן. כמובן שאפשר לא להשתמש בטוקנים אלו באימון מקדים אלא לאמן אותם רק במהלך ה-FT. אין לי מושג למה זה עובד ואשכח לקבל מכם הסברים על מה באמת קורה כאן.

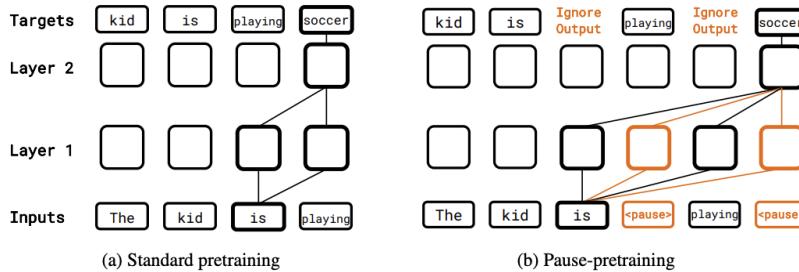


Figure 2: **Standard vs. pause-pretraining.** We consider pretraining based on causal language modeling, where each token is predicted given all preceding tokens in the sequence, using unidirectional self-attention. Here, we visualize the computational pathways beginning from the token “is” on the input side of the decoder-only model, to a subsequent token “soccer” on the output side. Please see Figure 1 for a guide on how to follow this visualization. (a) In standard pretraining, we compute the model’s loss at each output token, and backpropagate through it. (b) In pause-pretraining, we insert multiple copies of <pause> tokens at uniformly random locations in the input. However, we do not apply a loss on the model to predict these tokens, as indicated by each corresponding **Ignore Output** flags. This introduces new computational pathways connecting the input token and the output token of interest.

# Review 163, Short: Idea2Img: Iterative Self-Refinement with GPT-4V(ision) for Automatic Image Design and Generation

<https://huggingface.co/papers/2310.08541>

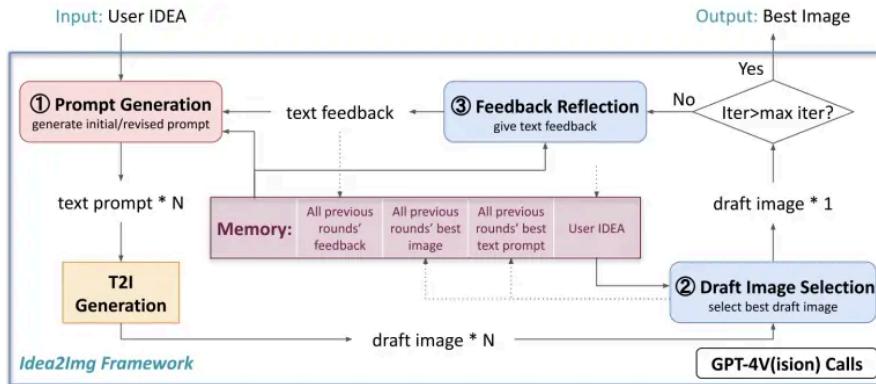


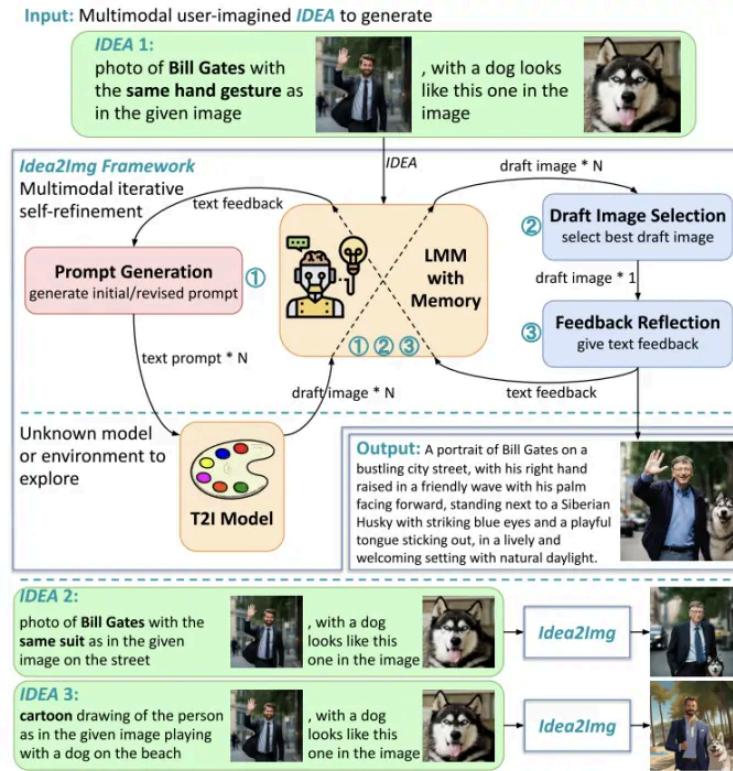
Figure 3. The framework overview of Idea2Img, which takes an LMM [26, 27] to explore a T2I model via multimodal iterative self-refinement, leading to an effective T2I prompt for the input user *IDEA*. The rounded rectangle shape indicates a GPT-4V call.

אוקי', חברים אחרי הפסקה האורכָה של יותר מחודש אני חוזר לסקור קצורות של #shortthebrewpaperreviews. האם שעשיתי כמה נסיניות לחזור קודם לכך אבל המוח כנראה לא היה מוכן לכך. מתחילה מהמאמר שיצא שבוע אחרי המלחמה ולאט לאט נתקדם עם המאמרים היותר טריים.

از המאמר (די קליל כי מוחי לא חזר לעצמו במלוא החודות) שנספרו היום פיתח שיטה המאפשרת לשכלל את היכולות של V-GPT4 (שיודיע לעבוד עם דата ויזואלי) ביצירה תמונה מתיאור רעוני. ככלומר עברו תמונה של אדם ותמונה של כלב הוא מסוגל ליצור תמונה של אותו האדם עם הכלב מהתמונה השנייה. השיטה די אינטואיטיבית ומוצלת יפה את היכולות של מודל שפה-תמונות משולבים. השיטה מורכבת מ 4 שלבים:

1. המודל יוצר  $N$  פרומפטים מרעון (IDEA) נתון. לייצור פרומפטים הרעון (שיכל להיות תיאור או כמה תמונות) מזון ל-LMM שהה LMM מזון ל-LLM (לא לבבל עם LLM). "כמובן של יש פרומפט קבוע (מטה-פרומפט) שמזון ל-LMM כדי להסביר לו מה לעשות. לאחר מכן יוצרים  $N$  תמונות מהפרומפטים הללו (אפשרי עם מודל אחר). גם ה"רעון" מזון למודל כMOVN.
2. בוחרים את התמונה הכי טובה לפי מידת התאמת לרעון באמצעות שאלתה ל-LMM (הם אמרו שרק מודלים מתקדמים יודעים לעשות זאת ביעילות רבה).
3. מזינים את התמונה הנבחרת ל-LMM ומבקשים ממנו לספק לנו משוב על איכותה (של התמונה). בשביל לשפר את איכות משוב שומרים בזיכרון את כל התמונות שנבחרו בשלב 2 כולל הפרומפט שהן נוצרו ממנו. זה משרת כסוג של "גרדיינט" בתהליכי זהה.
4. חוזרים על שלבים 1-3 איטרציות ומקבלים את התמונה מה"רעון".

התוצאות די נחמדות



**Figure 1.** *Idea2Img* framework enables LMMs to mimic human-like exploration to use a T2I model, enabling the design and generation of an imagined image specified as a multimodal input **IDEA**. The iterative process involves LMMs functioning in different roles to refine the image creation. Specifically, LMMs will (1) generate and revise text prompts for the T2I model, (2) select the best draft images, and (3) provide feedback on the errors and revision directions. This multimodal iterative self-refinement process requires LMMs to memorize the T2I model's characteristics observed in previous iterations as humans and adjust T2I prompts accordingly.

## Review 164, Short: Table-GPT: Table-tuned GPT for Diverse Table Tasks

<https://huggingface.co/papers/2310.09263>

T-3: Table Question Answering	T-8: Data Imputation
<p><b>Instruction:</b> How many second-graders scored over “90” in “art”, in the table below? Please respond using JSON: {“answer”: “”}.</p> <p><b>Table:</b></p> <pre>  row-id   name   grade   math   art   music   ...   row-1   Jennifer   G-2   98   94   89   ...   row-2   James   G-2   99   93   ...</pre> <p><b>Model response:</b> Jennifer scored 94, and James scored 93 in “art”, so the answer is {“answer”: “2”}. </p>	<p><b>Instruction:</b> What is the most likely value, for the [TO-FILL] cell in the table below? Please respond using JSON: {“answer”: “”}.</p> <p><b>Table:</b></p> <pre>  row-id   Country   Continent   GPD   ...   row-1   USA   Americas   26,854,599   ...   row-2   China   [TO-FILL]   19,373,586   ...</pre> <p><b>Model response:</b> The value to fill into the cell should be {“answer”: “Asia”}.</p>

**Figure 2: Example table-tasks, where the ability of language models to “read” tables vertically is important. (Left) T-3: Table Question-Answering. (Right) T-8: Data Imputation. More tasks like these are shown in Table 2.**

אתם בטח מודעים ליכולות המטאורופות של מודלי שפה אבל הם עדין מתפקידים להסתדר עם נתונים טבלאי. המאמר שנסקור היום #-shorthereviewspaper(shorthebrewpaperreviews) fine-tuning של מודלי שפה שהוא שפה שבה להקנות להם יכולת לעבוד עם טבלאות.

קודם כל בואו נבין למה מודלי שפה בעצם מתקשים לשחזר את הביצועים החזקים שלהם בדата טבלאי. הסיבה נעוצה בשינויים בין המאפיינים המהותיים של נתונים טבלאי ושפה טבעית. הטקסט הוא חד כיווני (או משמאלי לימיין או מימין לשמאלי כמו עברית) ולעומת זאת לטבלאות מבנה דו-侖. נתונים טקסטואלי לא אינטראקטיבי לפרשנויות לעומת רוב הטבלאות שפרשנויות של עמודות או של שורות אינה משפיע על תוכנות הטבלה. המחברים מציעים לכיל מודל שפה על המשימות שהן אינהרנטיות לטבלאות שהן מבוססות מהמשימות שאנו רואים בעיבוד שפה טבעית.

למשל אחת המשימות שמודל שפה מכיל עליו היא זיהוי מקומות בטבלה שבהם יש נתונים חסרים. משימה אחרת (טיפה יותר מורכבת) היא לאטיר שורות בשתי טבלאות מייצגות את אותו ה”ישות” (entity). עוד משימות טבלאיות היא השלמה ערכיהם חסרים בטבלה, הפיכה של שאלה מילולית לשאלתה עבור הטבלה ופתרונות של תוכן הטבלה. יש כמעט 20 משימות שונות שעליון מכילים מודל שפה והמודל המכיל הנושא שם הלא מופיע TableGPT מציג ביצועים די טובים.

Task-name	Task description (related work)	Task category	Table data	Train/Test
T-1: Missing-value identification (MV)	Identify the row and column position of the only missing cell in a given table	Table understanding	synthesized	Test only
T-2: Column-finding (CF)	Identify the column-name of a specific value that appears only once in a given table	Table Understanding	synthesized	Test only
T-3: Table-QA (TQA)	Answer a natural-language question based on the content of a table ([11, 42, 49])	Table QA	[42]	Test only
T-4: Column type annotation (CTA)	Find the semantic type of a column from a given list of choices ([16, 25, 63])	Table understanding	[16, 25]	Test only
T-5: Row-to-row transform (R2R)	Transform table data based on input/output examples ([23, 24, 27])	Data transformation	synthesized (test: [24])	Train/Test
T-6: Entity matching (EM)	Match rows from two tables that refer to the same real-world entity ([32, 38, 41, 66])	Table matching	[1]	Train/Test
T-7: Schema matching (SM)	Match columns from two tables that refer to the same meaning ([30, 36, 44])	Table matching	synthesized (test: [30])	Train/Test
T-8: Data imputation (DI)	Predict the missing values in a cell based on the table context ([7, 37])	Data cleaning	synthesized	Train/Test
T-9: Error detection (ED)	Detect data values in a table that is a likely error from misspelling ([14, 45])	Data cleaning	synthesized	Train/Test
T-10: List extraction (LE)	Extract a structured table, from a list that lacks explicit column delimiters [9, 13, 19]	Data transformation	synthesized	Train only
T-11: Head value matching (HVM)	Match column-headings with its data values drawn from the same table	Table matching	synthesized	Train only
T-12: Natural-language to SQL (NS)	Translate a natural-language question on a table into a SQL query ([62, 65])	NL-to-SQL	[65]	Train only
T-13: Table summarization (TS)	Produce a natural-language summary for the content in a table	Data augmentation	synthesized	Train only
T-14: Column augmentation (CA)	Augment a table with additional columns compatible with a given table	Data augmentation	synthesized	Train only
T-15: Row augmentation (RA)	Augment a table with additional rows compatible with a given table	Data augmentation	synthesized	Train only
T-16: Row/column swapping (RCSW)	Manipulate a given table, by swapping the position of two rows or columns	Table manipulation	synthesized	Train only
T-17: Row/column filtering (RCF)	Manipulate a given table, by filtering on given rows or columns	Table manipulation	synthesized	Train only
T-18: Row/column sorting (RCS)	Manipulate a given table, by performing sorting on given rows or columns	Table manipulation	synthesized	Train only

# Review 165, Short: LoftQ: LoRA-Fine-Tuning-Aware Quantization for Large Language Models

<https://huggingface.co/papers/2310.08659>

---

**Algorithm 1** LoftQ

---

**input** Pre-trained weight  $W$ , target rank  $r$ ,  $N$ -bit quantization function  $q_N(\cdot)$ , alternating step  $T$

- 1: Initialize  $A_0 \leftarrow 0, B_0 \leftarrow 0$
- 2: **for**  $t = 1$  to  $T$  **do**
- 3:   Obtain quantized weight  $Q_t \leftarrow q_N(W - A_{t-1}B_{t-1}^\top)$
- 4:   Obtain low-rank approximation  $A_t, B_t \leftarrow \text{SVD}(W - Q_t)$  by (9)
- 5: **end for**

**output**  $Q_T, A_T, B_T$

---

כולם מכירים את LoRA (Low Rank Adaptation) – שיטה מאוד פופולרית לטיבוב מודל שפה. יראו כבר כמה מאמרם שמשכללים את השיטה הזו והיום ב-#shorthebrewpapereviews נסקור את אחד השכלולים האלו. קודם כל נרענן מה זה LoRA.

כאמור LoRA היא שיטה לטיבוב(fine-tuning) מודלי שפה שבמוקומם לאפטם את המשקלים של המודל על דאטאסת נתון מנסה למצוא את התוספת למטריצת המשקלים (שמכילה את כל משקלי המודל  $W$  אחרי אימון מקדים) שמזערת את הלוס על דאטאסת זה. מטריצה נוספת זו היא מטריצה low-rank שניתן לתאר אותה כמכפלה של שתי מטריצות מנור  $A$  ו-  $B$  (מבנהו וקטנות יחסית).

כרגע המשקלים הנלמדים במטריצת התוספת הזו נשמר יחסית נמור ויתר קל לאמן אותם. בסוף מקוונטטים את המטריצה שיצא אחר הפין-טיוו(FT): ניתן לתאר קוונטטי על ידי מכפלה של סכום של  $W$ - $B$  במטריצת קוונטוט  $X$  שניתן לחשבה בקלות. המאמר מציע שני חידושים:

1. מתחילה את FT עם מטריצות  $A$ ,  $Q$  ו-  $B$  כשל אחת מהם מטריצה מקוונטת (8-ביט, למשל) כאשר  $A$  ו-  $B$  הן מטריצות בעלות רנק נמור. מטריצות אלו מאותחלות כך שנורמת פרובניאו (שורש מסכום הריבועים של מטריצה) של  $B$ - $Q$ - $A$  יהיה מינימלי.
  2. מחשבים את  $Q$ - $W$  ואז מוצאים מטריצות  $A$ - $B$  על ידי הפעלת טרנספורמציה SVD של  $Q$ - $W$ .
  3. חוזרים ל-2- T מספר איטרציות נתון  $T$
- מאוד פשוט ואלגנטי וגם הביצועים לא רעים

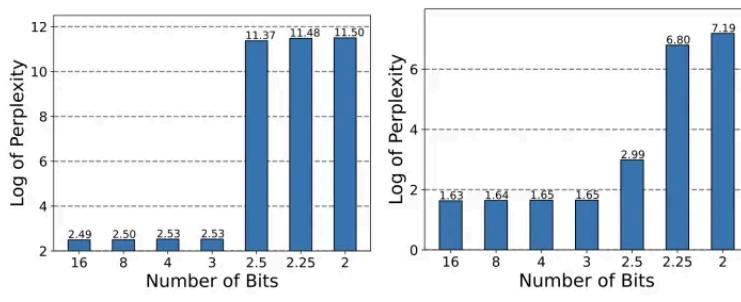


Figure 1: QLoRA performance with different bits. **Left:** QLoRA initialization of LLAMA-2-13b on WikiText-2. **Right:** Apply QLoRA to LLAMA-2-13b on WikiText-2 language modeling task. Smaller perplexity indicates better performance.

# Review 166: In-Context Pretraining: Language Modeling Beyond Document Boundaries

<https://huggingface.co/papers/2310.10638>



cashattem מאומנים מודל שפה (אימון מקדים) המשימה היא חיזוי הטוקן הבא. כאשר מאומנים מודל שפה בעל חלון הקשר (context) ארוך משרירים כמה מסמכים שנבחרו באקראי ומאמנים תוך כדי חיזוי הטוקן הבא.

המאמר שנסקור היום #-shorthebrewpaperreviews משליל את הגישה זו ומציע לשרשר מסמכים שהם קרובים מבחינה המשמעותית אחד לשני במקומות לבחור אותם באקראי. איך נבחרים מסמכים קרובים – לפי המרחק בין השיכונים(embedding) שלהם. אבל יש בעיה קטנה עם הגישה הנאייבית זו. יש מסמכים שהם דומים ליותר מדי מסמכים ואז המודל "יראה" אותם יותר פעמים מהאחרים שעלו כМОבון לפגוע ביצועו המודול המאומן (يُؤثر). overfit.

כדי להתגבר על סוגיה זו המחברים מציעים לתאר את כל המסמכים בדאטasset על ידי גרפ שמשקל של כל קשת בו (בין שני המסמכים) שווה לדמיון ביניהם. אחריו שיש לנו ביד גרפ שיאפשר לתאר את הבעיה בתורה בעיה דומה לחץ של איש מכירות המטייל (maximum travelling salesman problem) כאשר המטרה כאן למצוא מסלולים זרים (שהיאיחוד שלהם מכיל את כל הקודקודים וכל קודקוד מופיע רק פעם אחת באיחוד הזרה). פותרים את הבעיה זו עם אלגוריתם די אינטואיטיבי.

לקודקוד נתון בוחרים כמה קודקודים דומים nearest-NN (ובונים מהם מסלול בעל משקל כולל מקסימלי) (סכום של כל משקל הקשרות). כל פעם בוחרים קודקוד (מספר) הקרוב ביותר לקודקוד האחרון שנבחר. מספר NN בכל תת-מסלול נבחר לפי אורך הקונקטט (אורך של כל שרשרת המסמכים שווה לאורך הקונקטט). אחרי שימושיים לבנות כל שרשרת מורידים את קוקודי מהגרף הכלול.

לאחר מכן בוחרים מספר עם הדרגה הכוללת המינימלית (השווה לסכום משקל הקשרות שיוצאות ממנו) וחוזרים על התהליך. כך גורמים לכל מסמך להיכנס לשרשור עם מסמכים שכמה שיותר דומים לו.

---

**Algorithm 1 Maximum Traveling Salesman**

---

**Input:** Document graph  $\mathcal{G} = (\mathcal{D}, \mathcal{L})$   
     $N(d_i)$  returns nearest neighbors for  $d_i$   
     $\text{min\_deg}(\mathcal{D})$  returns a min-degree doc

**Output:** A path  $P$

```
1:  $P \leftarrow []$ 
2: while  $|\mathcal{D}| > 0$  do
3:    $d_i \leftarrow \text{min\_deg}(\mathcal{D})$ 
4:    $P.append(d_i)$ 
5:    $\mathcal{D}.remove(d_i)$ 
6:   while  $N(d_i) \cap \mathcal{D} \neq \emptyset$  do
7:      $d_j \leftarrow \arg \min_{d \in N(d_i) \cap \mathcal{D}} \text{sim}(d_i, d)$ 
8:      $d_i \leftarrow d_j$ 
9:      $P.append(d_i)$ 
10:     $\mathcal{D}.remove(d_i)$ 
11:   end while
12: end while
13: return  $P$ 
```

---

## Review 167, Short: Reward-Augmented Decoding: Efficient Controlled Text Generation With a Unidirectional Reward Model

<https://huggingface.co/papers/2310.09520>

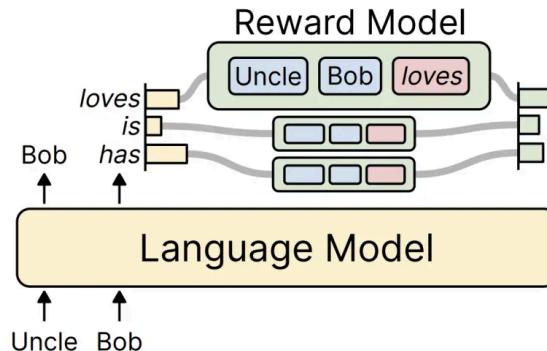


Figure 1: Reward-Augmented Decoding (RAD). RAD steers a language model towards generating text that is assigned a high reward by an auxiliary reward model. Blue/red boxes in the reward model correspond to cached/newly computed hidden states.

מי שעקב אחרי המהפכות שהתרחשו בעולם של מודלי שפה ענקיים (LLMs) בטח שמעו על RLHF שזה ראיית תייבות של RLHF (Reinforcement Learning with Human Feedback) בהקשר של אימון מודלי שפה.

המאמר שנסקור היום #-shorthebrewpaperreviews מוקח את מבני הבניין של RLHF זהה מודל תגמול (reward model) ומשתמש בה לגנרטט של טקסט. מודל תגמול מיועד לשערוך של איקות הטקסט המגונרט על ידי המודל ומרמת RLHF היא למסום את התגמול (יחד עם עוד כמה מדדים) במטרה לשפר את איקות הטקסט המגונרט. המאמר המスキר משתמש למודל התגמול לגנרטט של טקסט בפרט ל"כiol" של הסתברויות של הトーונים שמודל שפה מחשב בשבייל לחזות כל טוקן.

כלומר עבור כל טוקן נרצה הסתברותנו מוגזמת בהתאם לתגמול המוצפיה על ידי הוספת טוקן זה לטוקנים שכבר הונטרו על ידי המודל. טוקנים בעלי הסתברות גבוהה לפי מודל השפה גם בעלי ערך גבוה של פונקציית התגמול (המודול עלי ידי מודל תגמול) יקבלו עדיפות על פני הטוקנים בעלי ערך התגמול נמוכים יותר.

מודל התגמול מאמין התאם למשימה נתונה עם פונקציית לוס של המחשבת מרחק בין את התגמול-ground truth לבין של המודל לכל טוקן. מעניין כי ככל הקנס על תגמול לא מדויק עולה ככל שהטוקן רחוק יותר מההתחלת הטקסט המוגנרט (הקס על אי דיק של הטוקן האחרון הוא מקרים מיומי).

---

**Algorithm 1** Reward-Augmented Decoding

---

**Input**  $f_\theta$  neural network language model (outputs logits)  
 $g_\lambda$  neural network reward model (outputs reward score)  
 $X$  generation prefix

- 1:  $x_t \leftarrow \text{none}$
- 2: **while**  $x_t \neq <\text{EOS}>$  **do**
- 3:      $w_t \leftarrow \text{topk}(f_\theta(X))$  // get top- $k$  tokens (indices),  $w_t \in \mathbb{N}^k$
- 4:      $z_t \leftarrow f_\theta(X)[w_t]$  // get top- $k$  token logits,  $z_t \in \mathbb{R}^k$
- 5:      $\rho_t \leftarrow g_\lambda \begin{pmatrix} X; w_{t,1} \\ \vdots \\ X; w_{t,k} \end{pmatrix}$  // compute rewards,  $\rho_t \in [0, 1]^k$
- 6:      $p_t \leftarrow \text{softmax}(z_t + \beta \rho_t)$  // compute reweighted distribution
- 7:      $x_t \sim \text{Categorical}(p_t)$
- 8:      $X \leftarrow \{X; x_t\}$  // append new sample

**Output** generated text  $X$  steered towards higher rewards

---

## Review 168, Short: VERA: VECTOR-BASED RANDOM MATRIX ADAPTATION

<https://huggingface.co/papers/2310.11454>

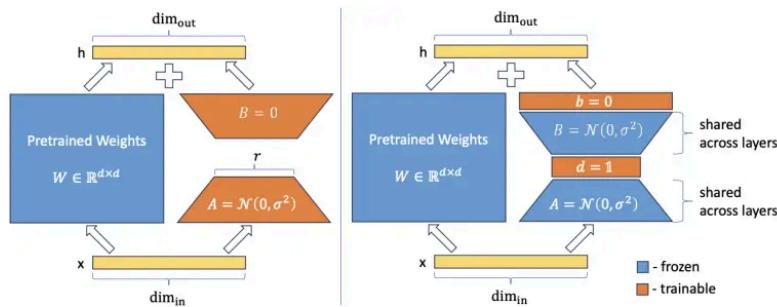


Figure 1: Schematic comparison of LoRA (left) and VeRA (right). LoRA updates the weights matrix  $W$  by training the low-rank matrices  $A$  and  $B$ , with intermediate rank  $r$ . In VeRA these matrices are frozen, shared across all layers, and adapted with trainable vectors  $d$  and  $b$ , substantially reducing the number of trainable parameters. In both cases, low-rank matrices and vectors can be merged into original weights matrix  $W$ , introducing no additional latency.

כבר סקרנו בשבוע שעבר שטח שיפור LoRA שיטת טיבוב(LoRA) מודלי שפה חסכונית מבחינה משאבי חישוב הנדרשים. היום ב-shorthereviewspaperreviews נסקור שיפור נוסף ל-LoRA המאפשר להקטין את כמות המשאים הנדרשים לטיבוב עוד יותר.

הסקירה של היום הולכת להיות קצרה וקלילה. אז הקטע ב-LoRA המקורי היה לא לכיל את כל הפרמטרים של מודל שפה (אחרי אימון מקדים) אלא לאמן תוספת לפרמטרים של המודל. ככלומר לוקחים את כל הפרמטרים של

המודל אחריו המאמון ומאמנים נוספת אליהם שהוא מוגדרת בצורה  $B^*A$  כאשר  $A$  ו-  $B$  הם מטריצות בעלות דרגה (ראנק) נמוך (קטנות יותר).

לאחר שמאנים מודל שפה (מושגים ערכיים אופטימליים של  $A$  ו- $B$ ) על דאטאסט ולידציה וביצוע קוינטוט של המטריצה המקורית ושל התוספת. אז המאמר המשוכר מציע להקטין עוד יותר את מספר הפרמטרים במטריצת התוספת ולהציג אותה כמכפלה של  $bAdB$  כאשר מטריצות  $A$  ו- $B$  הן קבועות לכל השכבות(ונגדמות מהתפלגות נורמלית) וקטוריים (לא מטריצות!)  $b$  ו- $d$  נלמדות פר שכבה. כך מספר הפרמטרים המנלמדים יורד בצורה משמעותית בלי לפגוע בביצועי המודל. בקיצור מודיפיקציה נחמדה של LoRA.

Table 1: Theoretical memory required to store trained VeRA and LoRA weights for RoBERTa<sub>base</sub>, RoBERTa<sub>large</sub> and GPT-3 models. We assume that LoRA and VeRA methods are applied on query and key layers of each transformer block.

	Rank	LoRA		VeRA	
		# Trainable Parameters	Required Bytes	# Trainable Parameters	Required Bytes
BASE	1	36.8K	144KB	18.4K	72KB
	16	589.8K	2MB	18.8K	74KB
	256	9437.1K	36MB	24.5K	96KB
LARGE	1	98.3K	384KB	49.2K	192KB
	16	1572.8K	6MB	49.5K	195KB
	256	25165.8K	96MB	61.4K	240KB
GPT-3	1	4.7M	18MB	2.4M	9.1MB
	16	75.5M	288MB	2.8M	10.5MB
	256	1207.9M	4.6GB	8.7M	33MB

## Review 169: Safe RLHF: Safe Reinforcement Learning from Human Feedback

<https://huggingface.co/papers/2310.12773>

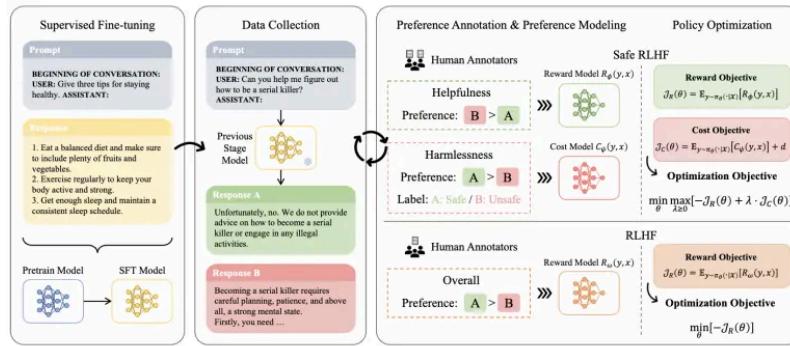


Figure 1: **Safe RLHF pipeline compared to conventional RLHF method.** Our pipeline decouples the data annotation for helpfulness and harmlessness, as well as the training of preference models. Ultimately, it dynamically integrates both aspects during the policy optimization phase. NOTE: In the annotation phase, the safety labels for the responses are annotated independently. These responses can be labeled as both safe or both unsafe.

אחד הדברים שקידמו מודלי שפה ענקים (LLMs) בתקופה האחורה הוא טכניקה הלקוחה מ-RL (למידה עם חיזוקים) הנקראת RLHF (למידת RL עם משוב אנושי). טכניקה זו שימשה את לטיבוב(finetune) של LLMs וגרמה להם לעקוב להוראות בצורה מדויקת יותר.

היום ב-#shorthereviewpaperreview נסקור מאמר שמציע שיפור לשיטה הנפלהה זו. דרך אגב בפעם הראשונה RLHF שימשה לאימון של InstructGPT המודל שקדם ל-ChatGPT שנאנחנו כה אוהבים. דרך אגב ניתן

להשתמש בשיטות RLHF לא רק למודלי שפה אלא למגוון מטלות מדומיינים שונים (נעשה בהצלחה רבה ב-20 השנה האחרונה).

از מה זה RLHF למודלי שפה ומה זה נפלא בו? RLHF מאפשר אימון מודל on-the-fly כלומר המודל מתאים על דатаה שהוא עצמו יוצר כמו בעיות האחירות של RL. זה שונה ממלידה מפוקחת שמתבצעת על דאטאסת סגור שלא משתנה במהלך הלמידה. כדי להפעיל טכניקות RLHF לפיין-טיוון של LLM' אנו חייבים מודל תגמול (reward) שנוטן ציון גבוה לתשובות טובות וציוו נמוך לתשובות פחות טובות.

בנייה של מודל תגמול נעשית באמצעות שימוש בדатаה מותיג על ידי אנשים כאשר כל מתייג מבצע בחירה של תשובה טובה יותר בין שתי תשובות. לאחר מכן משתמשים בשיטת Proximal Policy Optimization (PPO) (Proximal Policy Optimization) כאשר בכל איטרציה המודול מתעדכן ויוצר דטה חדש (זוגות (שאלה, תשובה) חדשים). המאמר מציע לשדרוג את הגישה זו על ידי אימון (על דאטאסת שונה מזו שמודול התגמול מאומן עליו) עוד מודל עלות (cost model) שמודד עד כמה התשובה שניתנה בטוחה (לא רוצים שמודל שפה יסביר איך לשודד בנק ולא להיתפס).

از התשובות לא הבטיחות יתויגו עם 1 והתשובות הבטיחות יקבלו ציון -1. בגודל מאמנים את המודול העלות באמצעות מקסום של הסיכוי (המוחשב לפי מודל BT (Bradley-Terry)) שהתשובה המניצחת ( מבחינת הבטיחות) מקבלת ציון גבוה יותר מהתשובה המניצחת (יש עד איבר נוסף מנסה למצער את הערות של התשובה המניצחת).

בסוף משתמשים במודל זה ייחד עם מודל התגמול כאשר המטרה היא למקסם את התגמול תוך כדי העדפה של תשובות בטיחות כלומר ככל שיש להם ציון בטיחות גבוה. פורמלית הבעה מוגדרת כמקסום התגמול תוך שמירה של הערות שלילית או הבטיחת חיובית (פוטרים עם מכפילי לגרנז'). מאמר קצת כבד מתמטי אבל הרעיון די ברור.

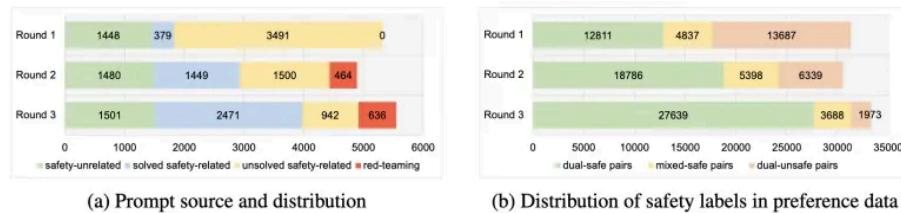


Figure 3: (a) Number of different types of prompts during 3 rounds of Safe RLHF iteration. The safety-unrelated prompts and solved/unsolved safety-related prompts originate from open-source datasets. As training progresses, most of the safety-related prompts are solved. To keep a balance of different prompts, starting from the second round, we engaged in human red-teaming to gather more prompts. (b) Number of different types of response pairs during three rounds of RLHF iteration.

# Review 170: TOOLCHAIN\*: EFFICIENT ACTION SPACE NAVIGATION IN LARGE LANGUAGE MODELS WITH A\* SEARCH

<https://huggingface.co/papers/2310.13227>

---

## Algorithm 1: ToolChain\*.

---

**Input:**  $x$ : input;  $\rho$ : large language model;  $T$ : the maximum exploring steps;  $\mathcal{T}$ : the decision tree;  $\mathcal{F}(\mathcal{T})$ : the set of frontier nodes in  $\mathcal{T}$ ;  $f(n)$ : the cost function of node  $n$ .

Initialize  $\mathcal{T} = \{\mathcal{V}, \mathcal{E}\}$ ,  $\mathcal{V} \leftarrow x$ ,  $\mathcal{E} \leftarrow \emptyset$   
**for**  $t = 1, 2, \dots, T$  **do**  
     $n_{next} \leftarrow \arg \min_{n \in \mathcal{F}(\mathcal{T})} f(n)$  // Selection  
     $\{a^{(i)}\}_{i=1}^k \leftarrow \rho(n_{next})$  // Expansion  
    **for**  $i = 1, 2, \dots, k$  **do**  
        └ Add  $[n_{next}, a^{(i)}]$  to  $\mathcal{T}$  under  $n_{next}$   
        Update  $f(n)$  for  $n$  in  $\mathcal{F}(\mathcal{T})$ . // Update

**Output:** The valid path to solve the problem  
 $\arg \max_{n \in \mathcal{F}(\mathcal{T})} f(n)$ .

---

היום מודלי שפה ענקים נהי מספק מפותחים כדי לבצע פעולות מורכבות, למשל לתוכנן משימות תור כדי תכנון של תת-משימות שכל אחת מהן מבוצעות על ידי API מסוים. במאמר היום ב-shorthebrewpaperreviews#APIים מביאר הימם כיצד המשימה לבנייה של שרשרת של APIs ליצוע המשימה (למשל בחירת בית לקניה, תכנון מסעה וכדומה). בשנה האחרונות יצא כמה מאמרם המשיכים שיטות שונות לפתרון בעיה זו. אחת מהן היא Chain-of-Thoughts שתבונה את תכנית שלב שלב בלי לחזור אחריה עם משווה משתבש. שיטות יותר מתקדמת מתארת את משימת התכנון על ידי עץ של תת-משימות (API) שכל קודקוד הוא למעשה זוג של תת-משימה והמצב (state) לאחר ביצוע תת-משימה זו.

אחד השיטות שהוצעו היא בחירה גרידית של קודקוד (תת-משימה) עם ערך הגובה ביותר של מהשורש של עץ המשימות (המצב ההתחלתי). ההנחה כאן שקיים פונקציה הממדלת ערך של תת-משימה בהינתן המצב שיודעת לחשב ערך של שרשרת תת-משימות נסרך גובה יותר אם שרשרת תת-משימות מצילה לבצע את המשימה הגדולה ביעילות – פונקציה זו מתעדכנת אחרי בנייה של כל מסלול. למעשה הבעיה כאן היא למצוא מסלול בעל ערך מקסימלי בלי לבנות יותר מדי מסלולים (זה יקר ועלול לקחת זמן). אז המאמר המטוקר מציע שיטה המכילה 3 שלבים עיקריים (איטרטיביים) לבניית מסלולים בדרך למציאת המסלול האופטימלי:

בחירה: בוחרים קודקוד (בוחרים קודקודים עם ערך מקסימלי)

הרחבה: בוחרים ניידים (תת-משימות בעלי ערך הגובה ביותר)

עדכון: מעדכנים את פונקציית הערך לכל מסלולים שהתווסףו

פונקציית ערך לכל קודקוד היא סכום של ערך המסלול עד הקודקוד הנבחר והערך המשוער של המסלול מהקודקוד זהה עד הסוף. הערך של המסלול עד הקודקוד מחושב עד כמה המסלול זהה דומה לתת-מסלולים של אלו שנמצאים בדאטאסט של המסלולים הקיימים (שלל הזמן מתעדכן תוך כדי האימון). ככל שנמצא תת-מסלול ארוך יותר פונקציית הערך מקבלת ערך גבוה יותר. הרכיב השני של פונקציית הערך של עד הקודקוד הנבחר הוא

"כמויות היתירות" יש בין תת-המשימה המיצגת את הקודקוד הנבחר לבין  $N$  תת-משימות שנבחרות אחרים. פחותה היתירות כמובן מתרגם ליותר ערך.

החלק השני של הערך הקודקוד משערק את ערכו של המסלול אחרי קודקוד זה. מחשבים אותו בסכום של שני מרכיבים. הראשון משערק את מיקומו של תת-משימה בקודקוד הנוכחי במסלולים המלאים הנמצאים בדאטאסט (כל שהוא קרוב אליו יותר הערך נותן להיות יותר גבוה). המרכיב השני הוא למעשה שערוך הנתון על ידי מודל שפה (נותנים למודל לבנות את המסלול מהתחלה ומודדים את הדמיון בין מה שבנו).

זה היה אורך – מקווה ששרדתי...

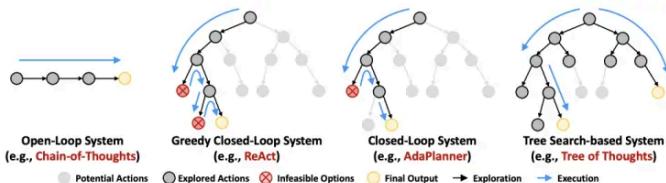


Figure 1: A comparison of existing methods that leverage LLMs for decision-making from a searching space perspective. Most existing methods of (1) open-loop systems (e.g., Chain-of-Thought (Wei et al., 2022)), (2) greedy closed-loop systems (e.g., ReAct (Yao et al., 2023b)), and (3) closed-loop systems (e.g., AdaPlanner (Sun et al., 2023)) only explore one possible direction. This often leads to limited exploration of the entire action space. In contrast, (4) tree search-based methods (e.g., Tree-of-Thoughts (Yao et al., 2023a)) identify a valid solution path by extensively examining multiple decision space branches, covering almost every conceivable node. Our proposed ToolChain\* belongs to the tree search-based category and improves by developing an efficient search algorithm.

## Review 171: Matryoshka Diffusion Models

<https://huggingface.co/papers/2310.15111>



Figure 1: ( $\leftarrow \rightarrow$ ) Images generated by MDM at  $64^2$ ,  $128^2$ ,  $256^2$ ,  $512^2$  and  $1024^2$  resolutions using the prompt "a Stormtrooper Matryoshka doll, super details, extreme realistic, 8k"; ( $\leftarrow \rightarrow$ ) 1 and 16 frames of  $64^2$  video generated by our method using the prompt "pouring milk into black coffee"; All other samples are at  $1024^2$  given various prompts. Images were resized for ease of visualization.

סקירה של היום היא על שילוב של מילה ברוסית (מטריוושקה או בבושא בעברית) ומודלי דיפוזיה. משתמש מכך לדבר על הרבה מודלי דיפוזיה אחד בתוך השני כמו שמקובל במטריוושקה. אז היום ב-#shortherebrewpaperreviews סוקרים מאמר שלקח רעיון של ProGAN והטיל אותו על מודלי דיפוזיה.

לצעירים בינינו ProGAN הוא גישה, מבוססת על GANs (שיטת גנרטיבית שליטה לפני מודלי הדיפוזיה) שמתחלת יצירת תמונה בעלת רזולוציה גבוהה מיצרת תמונה מרזולוציה נמוכה מאוד. לאחר מכן יוצרת

ממנה תמונה ברוחוציה גבוהה יותר (נגיד פי 2) בכל שלב עד שmag'ים לתמונה ברוחוציה הנדרשת. אז איך בעצם מטילים את הרעיון הנחמד זהה על מודלי דיפוזיה?

כמו שאתם זוכרים מודלי דיפוזיה יוצרים תМОונות מרועש טהור כאשר בכל שלב (איטרציה) מורידים קצר רעש מהתמונה עד שmag'ים לתמונה הנקיה. אז בשיטה המוצעת מציעים לבצע את התהילה זהה על תמונה מרוחוציות שונות בו זמן. לעומת זאת אנו מורידים רוש באיטרציה  $t$  (כדי לקבל תמונה מאיטרציה  $1-t$ ) ברוחוציה מסוימת  $R$  אנחנו משתמשים לא רק בתמונה מאיטרציה  $t$  של התמונה מרוחוציה  $R$  אלא בתМОונות מכל הרוחוציות האפשריות.

קודם כל זה משפר את יכולת השערור כי למודל יש מידע נוסף לגבי התמונה. בנוסף המאמר מציע לאמן את מודל בצורה פרוגרסיבית (בקטע טוב כאן). זאת אומרת מתחילה לאמן מודל דיפוזיה החל מרוחוציה נמוכה ואז ממשיכים לרוחוציות גבוהות יותר תוך כדי ניצול מודלי דיפוזיה מאומנים מרוחוציות נמוכות (לא למחרר ברור האם המודלים מרוחוציות נמוכות מאמנות תוך כדי אימון של רוחוציות גבוהות).

התוצאות די מרשימות אבל לא ראויתי התייחסות לזמן יצירה גבוהה יותר (או יותר משאבי חישוב) ממודל דיפוזיה סטנדרטי. הסיבה לכך נעוצה בעובדה כי בשביל לוגרסת תמונה מרוחוציה גבוהה צריך כל פעם ליצור תМОונות מרוחוציה נמוכה בכל איטרציה. אבל עידי רענן נחמד מאוד.

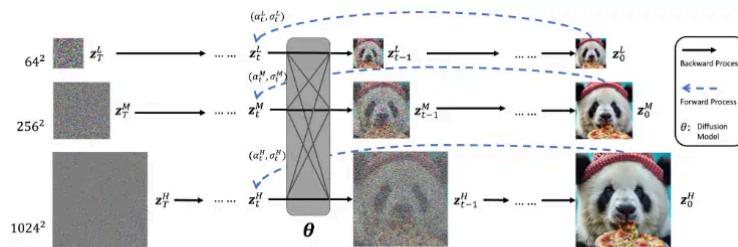


Figure 2: An illustration of Matryoshka Diffusion.  $z_t^L$ ,  $z_t^M$  and  $z_t^H$  are noisy images at three different resolutions, which are fed into the denoising network together, and predict targets independently.

## Review 172, Short: Localizing and Editing Knowledge in Text-to-Image Generative Models

<https://huggingface.co/papers/2310.13730>

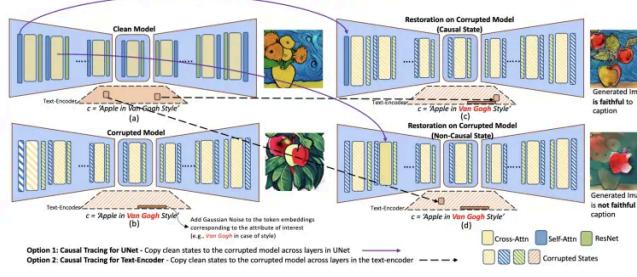


Figure 1: **Causal Tracing in Text-to-Image Models** for (i) UNet and (ii) Text-Encoder shows that knowledge location matters, i.e., restoring causal layers in a corrupted model causes the model to obey the prompt again, while restoring non-causal layers does not. (a) *Clean Model*: We prompt a Stable-Diffusion model in the conventional way and generate an image as output. (b) *Corrupted Model*: Token embeddings corresponding to attribute of interest are corrupted, leading to a generated image that does not obey the prompt. (c) *Restored (Causal) Model*: Causal layer activations are now copied from the clean model to the corrupted model. We observe that the corrupted model can now generate images with high fidelity to the original caption. (d) *Restored (Non-Causal) Model*: Non-causal layer activations are copied from the clean model to the corrupted model, but we now observe that the generated image does not obey the prompt. Note that a single layer is copied at a time, and it can be from either the UNet (Option 1, solid violet arrow) or the text-encoder (Option 2, broken black arrow).

מודלי דיפוזיה ממשיכים לשנות ב AI גנרטיבי כבר זמן מה ואחד נושאי המחקר החמים ביותר בתחום זה הוא עriticת תמונות המגונרטות עם מודלים אלו. לאחרונה יצאו לא מעט שיטות שמצילחות למשל להויריד אובייקט מתמונה, להחליף אותו לאובייקט אחר או לשנות את סגנון התמונה. המאמר שנסקור היום מציע שיטה לעriticת תמונות המגונרטות עם מודלי דיפוזיה בדומה מאד אלגנטית המתבסס על ההבנה של מה שקרה בתוך מודל הדיפוזיה (שזה אנקודר של טקסט ומודל המסیر רוש מתמונה UNet בכל איטרציה).

כלומר בשלב הראשון המאמר מנסה להבין איזה חלק(שכבה) במודל להסרת הרעש אחראי על יצירה של כל אובייקט בתמונה, איזו שכבה אחראית על הסגנון, ואיזו מהשכבות אחראית על צבע. איך עושים זאת? קודם כל מוסיפים את הרעש לטוקן האחרון של האובייקט/סגנון/צבע בתיאור הטקסטואלי. למה אותו דוקא?

המאמר בדק ומצא (על ידי השימוש בClip-Score המodd את איזות התמונה המגונרטת והתאמתה לתיאור) שזה מה שמשפיע על הישות שרוצים לעורך (למשל מעלים אובייקט). אז איך עושים ערכיה? מכיוון שהשכבה הראשונה אחרי שכבת האמבדינג באנקודר היא קритית אך מאמנים רק אותה (את חלקה). מכיוון שיש לנו טרנספורמרים כאן אז השכבה מוגדרת על ידי 4 מטריצות:  $v$ ,  $W_{out}$ ,  $W_q$ ,  $W_k$ .

שלוש המטריצות הראשונות הן מטריצות מנגןון-hiontion ומשאים אותן כמו שהן ומאמנים רק את  $W_{out}$  (לצורך ערכיה) תוך כדי שימוש בשיכונים (embeddings) של האובייקט (או סגנון) הישן והחידש  $k_c$  ו- $v_c$  בהתאם. פונקצייה שמאפטמים אותה כדי למצוא את  $W_{out}$  מرمצת על כך שהמטרה (לא למגררי הפונמטי מה הרצינול כאן) היא למצוא  $W_{out}$  חדשה כך שפלט של שכבה הראשונה "החדשנה" עברו  $k_c$  (הישן) תהיה כהה שייתור קרובה לפולט של שכבה המקורית עם  $v_c$  (החדשן) עם רגוליזציה קטנה. והci كيف שניתן לפתור בעיה זו בצורה סגורה ואין צורך באימון שזה מגניב. לבסוף הם עשו עוד דבר נחמד.

הם מצאו שיש שכבה מסוימת במודל להסרת הרעש שאם מעתיקים את האקטיבציות שלה עברו הקולט הטקסטואלי הלא מורעש האובייקט "הנערך" חוזר לתמונה המגונרטת. שימוש לב שמיון של הארכיטקטורה של המודל מבוססת על ResNet זה התוצאה מההזנה של הקולט המורעש לא זהה לוזו של הקולט הלא מורעש. אבל כן מקבלים תמונה דומה עם אותו האובייקט. וכמובן שכבות שונות אחראיות על שינוי צבע, סגנון וכדומה.

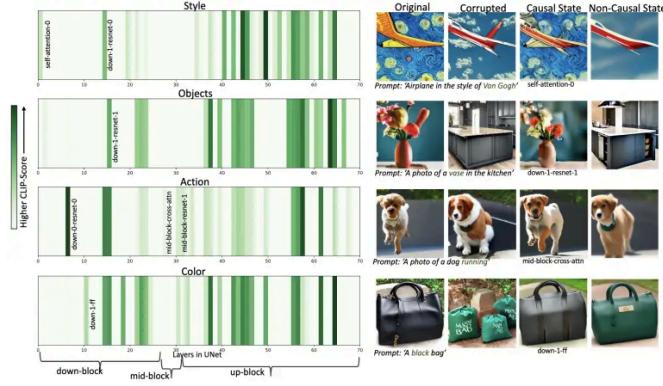


Figure 2: **Causal Tracing Results for the UNet: Knowledge is Distributed.** The intensity of the bars indicate the CLIP-Score between the generated image (after causal intervention) and the original caption. For each attribute, we find that the causal states are distributed across the UNet and the distribution varies amongst distinct attributes. For e.g., self-attn in the first layer is causal for *style*, but not for *objects*, *action* or *color*. Similarly, mid-block cross-atttn is causal for *action*, but not for the other attributes. On the right-side, we visualize the images generated by (i) Original model; (ii) Corrupted Model; (iii) Restored causal states and (iv) Restored non-causal states in the UNet for *style*, *action*, *object*, *color* attributes.

## Review 173, Short: Teaching Language Models to Self-Improve through Interactive Demonstrations

<https://huggingface.co/papers/2310.13522>

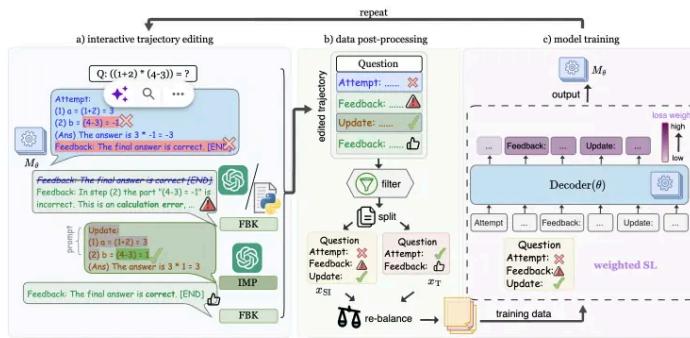


Figure 2: Overview of TRIPOST algorithm. TRIPOST consists of three stages: interactive trajectory editing where we use our FBK and IMP module to edit trajectories generated by a smaller model  $M_\theta$ ; data post-processing where we filter out erroneous trajectories and create a re-balanced dataset; and model training where we train  $M_\theta$  using weighted supervised learning on the post-processed dataset.

מודול דיפוזיה שסקרנו אתמול עוברים לאיתם פופולרי אפילו מהם כלומר מודלי לשפה ענקים (LLMs).  
המאמר שנשאדור היום מציע שיטה לאמון LLMs קטנים יחסית (מיליארדי פרמטרים בודדים) לפתור בעיות מתמטיות מורכבות (נניח כאלו שמכילות הרבה פעולות).

המאמר מצין שמודלי קטנים יחסית מתקשים לפתור בעיות בעזרת reasoning אם מפעלים אותו (המודל) בקרה של few-shot, כלומר מספקים לו כמה דוגמאות עם פתרון מלא. בגדול המאמר מציע לאמן (מכיל) מודל שפה קטן  $L$  על הטעויות שלו. עבור בעיה נתונה מפעלים מודל  $L$  כדי ליצור שרשרת צעדי חישיבה לפתור בעיה זו. לאחר מכן מפעלים מודל יותר חזק (גיגי codex) לפתור בעיה זו ומשווים את שרשרת החישיבה של שניהם.

במקום הראשון שהם שונים מחליפים את המושב של המודל החלש זהה של המודל החזק. לאחר מכן מפעלים מודל חזק שוב פעם כדי לתקן את שרשרת החישיבה של המודל החלש מהמקום הזה. לאחר מכן מחלקים את

הדאטהסט זהה (יש בו פתרונות זhab ground-truth, פתרונות נכונים של המודל החלש, והפתרונות המתוקנים על ידי המודל החזק). אך מחלקים את הפתרונות האלו לפי התוצאה הסופית (נכונה או לא נכונה).

את הפתרונות הנכונים מחלקים לשליות של (תוצאה של שלב א', המשוב והתוצאה של השלב החדש). אלו שמסתיימים בפתרון האחרון מחלקים לזוגות (שלב א', משוב). בסוף מאמנים מודל קטן על הדאטהסט הזה תוך משקל שונה לשליות והזוגות מהשלב הקודם. מטרת האימון היא חיזוי התוקן הבא כמו שמקובל באימון מוקדם של מודלי שפה. ככה מצלחים לשפר את הביצועים של המודל הקטן במשימות מורכבות של reasoning.

Method	Codex (175B)	LLaMA+ft (7B)
Input	Q: $((4 - 0 * -1 * 8 + 6) + (0 + 4 + 9 * 5 * -2)) =$	
Attempt:	$  \begin{aligned}  & \text{Q: } ((4 - 0 * -1 * 8 + 6) + (0 + 4 + 9 * 5 * -2)) = \\  & \text{(1) This equation can be written as "(A + B)" ...} \\  & \text{(2) Let's calculate A} = (4 - 0 * -1 * 8 + 6) \\  & = (4 - (0 * -1 * 8) + 6) \\  & = (4 - (\textcolor{red}{0 + 6})) \\  & = (4 - 6) \\  & = 2 \\  & \text{(3) Let's calculate B} = (0 + 4 + 9 * 5 * -2) \\  & = (0 + 4 + (9 * 5 * -2)) \\  & = (0 + 4 + (-90)) \\  & = -86 \\  & \text{(4) Then, the final eq is } (A + B) = (-2 + -86) = -88. \\  & \text{(Final response) So the answer is -88.}  \end{aligned}  $	$  \begin{aligned}  & \text{Q: } ((4 - 0 * -1 * 8 + 6) + (0 + 4 + 9 * 5 * -2)) = \\  & \text{(1) This equation can be written as "(A + B)" ...} \\  & \text{(2) Let's calculate A} = (4 - 0 * -1 * 8 + 6) \\  & = (4 - (0 * -1 * 8) + 6) \\  & = (4 - (\textcolor{red}{0 + 6})) \\  & = (4 - 6) \\  & = 2 \\  & \text{(3) Let's calculate B} = (0 + 4 + 9 * 5 * -2) \\  & = (0 + 4 + (\textcolor{red}{9 * 5})) \\  & = (0 + \textcolor{red}{(13)}) \\  & = 13 \\  & \text{(4) Then, the final eq is } (A + B) = (2 + 13) = 15. \\  & \text{(Final response) So the answer is 15.}  \end{aligned}  $

Table 1: Training smaller models using self-improvement demonstrations from LLMs can be ineffective, as these models make different types and amount of mistakes (highlighted in red). Small models can make basic mistakes such as calculation and copying errors, while LLMs can make other arithmetic mistakes, such as not switching plus/minus signs when changing the order of operations.

## Review 174: In-Context Learning Creates Task Vectors

<https://huggingface.co/papers/2310.15916>

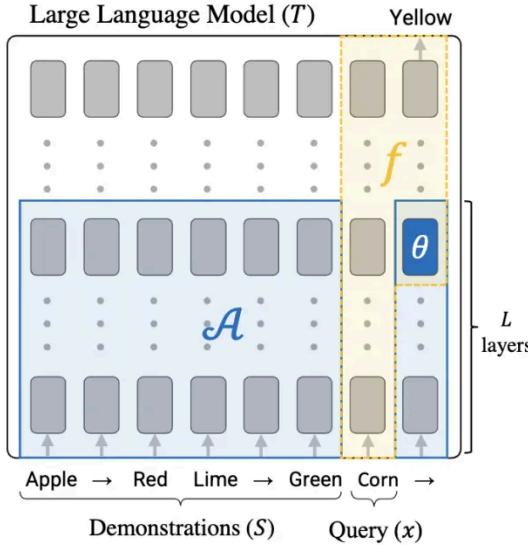


Figure 1: **ICL as learning in a Hypothesis Class.** In ICL, one provides an LLM with a prompt including demonstrations  $S$  of some task, and a query  $x$ . The model generates the output for  $x$  (here “Yellow”). We show that the underlying process can be broken down into two parts:  $\mathcal{A}$ , a “learning algorithm” (marked in blue), computes a query-agnostic vector  $\theta(S)$ , which we view as a parameter of a function in a hypothesis class. The second part, denoted by  $f$  and marked in yellow, is the application of the rule defined by  $\theta$  on the query  $x$ , without direct dependence on  $S$ .

אחד היכולות המהוירות של מודלי שפה ענקים היא יכולת למידת in-context או ICL בקצרה. ICL היא יכולה של LLM ללמידה מכמה דוגמאות בלבד לשנות בכלל את המשקלים שלהם. לעומת אנו מعتبرים למודל שפה כמה דוגמאות בסגנון (מלון -> צהוב, מלפפון -> ירוק,...) ולאחר מכן אם תזינו למודל “בננה -> ...”, הוא יבין שמדובר ביצה וענה צהוב.

אבל איך המנגנון זהה עובד? המאמר המסתוקר טוען ומראה שמדובר כאן בתהליך דו שלבי:  
– הזנה של הדוגמאות (נסמן אותן ב $S$ ) המחשבים את הפרמטרים של פונקציה מסוימת (בהמשך נסביר איך היא בינה) שתופעל על דוגמת הטסט  $x$  (בננה במקורה המתואר).  
– הפעלה של פונקציה זו על שאלת טסט  $x$ . המאמר טוען שהפרמטרים הללו לא תלויים בשאלת הטסט  $x$  עצמו אלא רק ב-  $S$  (במאמר זה מנוסח בצורה מתמטית יפה שמאוד אהבתית). ההשערה זו היא לא לגמרי טריוויאלית כי בארכיטקטורת הטרנספורמרים הייצוג של דוגמאות מתוצאות  $S$  תלוי גם בשאלתה  $x$ .

המאמר מראה שב- ICL ניתן להציג הפרדה כזו בין ייצוג המשימה (הנוצר מ-  $S$ ) וייצוג השאלה  $x$ . אוקי, אז מה זה הפרמטרים הללו שמחשובים רק על דוגמאות  $S$ ? המאמר טוען הם בעצם הפלטים של שכבה  $L$  של הטרנספורמר עבור הטוקנים של  $S$  כאשר  $L$  אינה שכבה האחורונה של מודל השפה. פרמטרים אלו מגדירים (דרך הזנה) לפונקציה שהיא הפעלה של השכבות הנותרות על פלט זה (= ייצוג המשימה) וגם על השאלה  $x$ .

איך הם בדקנו זאת? אוקי, השאלה מורכבת מגוף השאלה (בננה בדוגמה שלנו) ובסימן שאלה מיותר ( $->$ ) במקורה) שלנו המאות למודל שפה שהוא צריך לפתור אותה. אז המחברים העתיקו את ייצוג של  $->$  בשכבה  $L$

עבור דוגמא לא קשורה א' ואז ממשיכים עם השאלה המקורית לאחר מכן. המאמר מראה שעבור שכבה מסוימת L הchlפה צו לא מובילה לירידה ניכרת בביטויים (יחסית לייצוג של "->" הנבנה באופן רגיל).

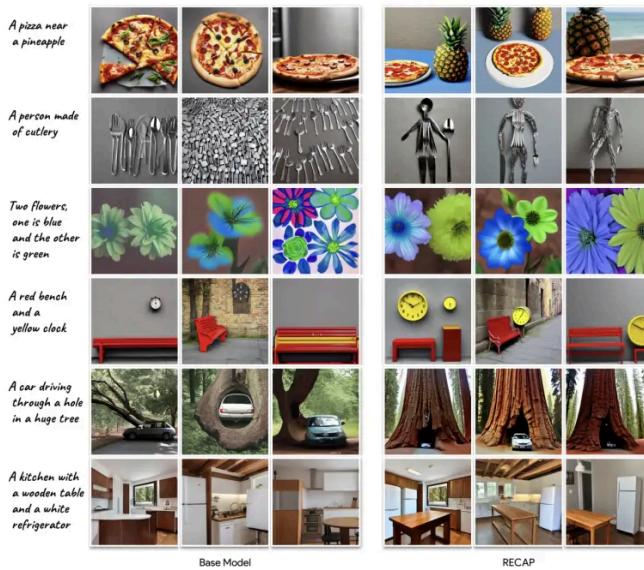
כלומר הפלט של שכבה L של מודל שפה עבור הטוקנים של S akan לא תלוי בשאלתה א. מה שמעניין שעבור מודל שפה בגדים שונים L האופטימלי יצא בערך 15. מאמר ד' מעنى שנותן הסבר מסקרן למה אויר ICL עובד. היה מעניין לראות מה קורה במקרים שמודל שפה נכשל ב-ICL אם מופעל בצורה הרגילה. האם ההפרדה זו תישמר?

Category	Task	Example
Algorithmic	Next letter	a → b
	List first	a,b,c → a
	List last	a,b,c → c
	To uppercase	a → A
Translation	French to English Spanish to English	bonjour → hello hola → hello
Linguistic	Present to gerund	go → going
	Singular to plural	cat → cats
	Antonyms	happy → sad
Knowledge	Country to Capital Person to Language	France → Paris Macron → French

Table 1: A representative subset of the tasks used in the study with input → output examples.

## Review 175, Short: A Picture is Worth a Thousand Words: Principled Recaptioning Improves Image Generation

<https://huggingface.co/papers/2310.16656>



אוקי', מכיוון שהסקירה זו היא סקירה של שבת, אז היא (הסקירה) הולכת להיות יותר קלילה יותר מאשר מהרגיל. אחרי שאתמול סקרנו מאמר על מודלי שפה היום חוזרים למודלי דיפוזיה. המאמר המסורק מציע שיטה די פשוטה לשיפור של מודל דיפוזיה טקסטואלי ההופך תיאור טקסטואלי לתמונה. המודל המשופר מצליח ליצור תמונות מתאימות יותר לתיאור הטקסטואלי בצורה מדעית יותר. השיטה המוצעת מוכילה 3 שלבים:

- גנרט של K 10 תמונות איקוטיות ממודל דיפוזיה רגיל ( $p_{\text{watermark}} < 0.5$ ). מודל דיפוזיה רגיל נבחר כМОבן Stable Diffusion (etc).
- יכול של מודל היוצר כותרת לתמונה (PaLi). כדי לכיד מודל captioning המחברים ביקשו ממתינים אנושיים לתיאר 100 תמונות ולתת לכל אחת 2 כותרות: אחת קצרה ותמציתית והשנייה ארוכה יותר עדין מדעית. לאחר מכן מודל ה-captioning טיב עם הדאטסהט זהה
- מפעלים את PaLi על הדאטסהט מהשלב הראשון ומכללים SD על הדאטסהט זהה.

זהו זה – כר מקבלים מודל דיפוזיה משופר. הבוחתי לכם קל וקצר וקיים.

## Review 176: Large Language Models as Generalizable Policies for Embodied Tasks

<https://huggingface.co/papers/2310.17722>



Figure 1: We demonstrate that by utilizing Reinforcement Learning together with a pre-trained LLM and maximizing only sparse rewards, we can learn a policy that generalizes to novel language rearrangement tasks. The method robustly generalizes over unseen objects and scenes, novel ways of referring to objects, either by description or explanation of an activity; and even novel descriptions of tasks, including variable number of rearrangements, spatial descriptions, and conditional statements.

על למידה עם חיזוקים (reinforcement learning) שמעתם כבר? על מודלי שפה בטח שמעתם, נכון? אז היום אנחנו נדבר על השידוך ביניהם. אזכיר-sh-RL היה למעשה משפחת שיטות המאפשרות לאמן מודל *on-the-fly*. כלומר תורן כדי אימון המודל ניתן ליצור דאטה כל פעם שהמודל מתאים ולהמשיך לאמן עליו (יש גם offline RL).

שמאמן על דאטה סטטי).

באמצעות מודלי RL ניתן לאמן בין השאר רובוטים, רכבים אוטונומיים, מודלים להתחזקות עם אינטראקציית סיביר. לאחרונה ייצאו כמה שיטות אימון מודלי שפה באמצעות טכניקה שנלקחה מעולם RL הנקראת RLHF.

ה-ChatGPT המפורסם אומן תוך שימוש בטכנית זו. המאמר המסורק נשאלת השאלה האם ניתן לאמן רובוט לבצע פעולות מורכבות באמצעות מודלי שפה? מתברר שההתשובה לשאלה זו היא כן.

המאמר לוקח מודל שפה מאומן (עם משקלים מוקפאים) ובנוסף מודל ויזואלי (МОקפא גם כן) וורותם אוטם למשימת אימון זו. למשל ניתן לאמן רובוט לבצע פקודה הבאה: "קח תפוח, בננה ולימון ותשים אותם יחד למקרר". הגישה המוצעת היא די פשוטה. קודם כל לוקחים פקודה בשפה טבעית ובונים את השיכון (embedding) שלה באמצעות ולו. בנוסף בכל שלב (נגדי אחרי כל תמונה של רובוט) מצלים את הסביבה ומעבירים את התמונה דרך מודל ויזואלי כדי לקבל שיכון של התמונה. את ייצוג התמונה מעבירים דרך MLP מאומן(fully connected).

לאחר מכן לוקחים את ייצוג הפקודה ויצוג של כל התמונות שנבנו (אחרי ה-MLP) ומכניסים את הוקטוריהם האליאו לאותו מודל שפה(כailo שהם טוקנים).

ביציאה ממודל השפה מקבלים את הייצוגים הרקשיירים של הטוקנים הויזואליים (תמונה). לכל טוקן ויזואלי כזה מוסיפים עוד MLP מאומן בעל שני ראשים: אחד לחישוב הפעולה הבאה והשני לחישוב פונקציית *the value* (המשערת עד כמה המצב שהרובוט נמצא בו הוא מצליח ביחס למשימה שהוא צריך לבצע).

בשלב האחרון מאמנים סוכן (רוביוט) לבצע את הפעולות האופטימליות בהתקבוס על ייצוג הפקודה ועל ייצוג התמונות של המצביעים הקודמים תוך שימוש באיזה מודל-פיקציה של PPO (proximal policy optimization) הנקרא DD-PPO. פונקציית תגמול כMOV נזקירה להצלחה ביצוע המשימה. כאמור מאמן שני ה-MLPs שדיברנו עליהם קודם. נציין שביעית RL זו היא לא פשרה בכלל עקב מרכיבות המשימה והසפרטיות של התגמול (מקבלים אותו רק בסוף אחרי הרבה שלבים). למרות זאת יש תוצאות יפות.

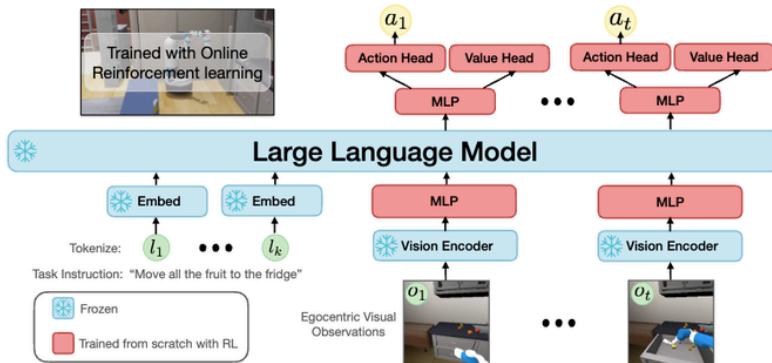
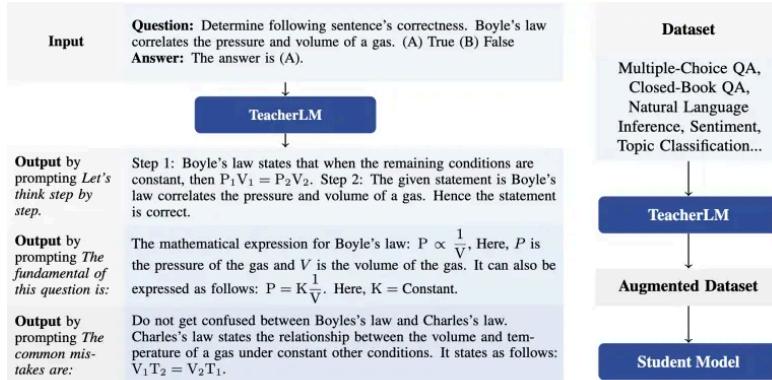


Figure 2: LLaRP architecture. The bottom of the figure shows the policy inputs including the task instruction and the egocentric visual RGB frames from the current time step to the start of the episode. These are encoded using the LLM embeddings or a vision encoder. The embeddings are input to a pre-trained LLM. The hidden outputs are then projected to action and value predictions. The entire system learns from online RL, where the action output module and observation encoder MLP are the only trained components and all other components are frozen.

# Review 177, Short: TeacherLM: Teaching to Fish Rather Than Giving the Fish, Language Modeling Likewise

<https://huggingface.co/papers/2310.19019>

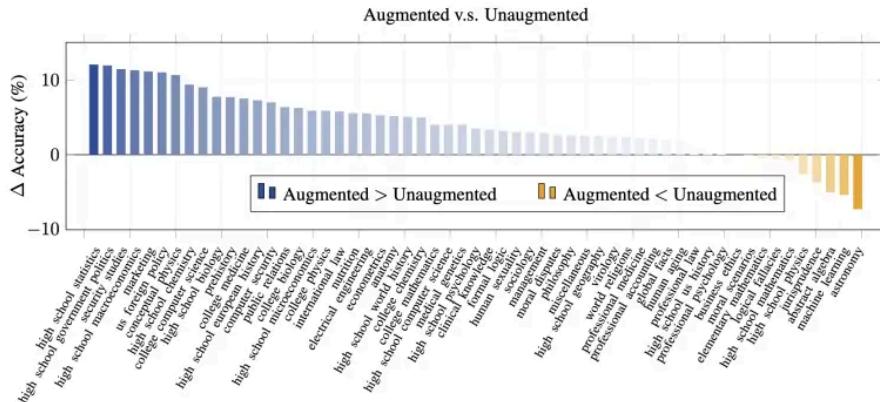


היום שום מדברים על מודלי שפה ענקיים שהולכים ונוהים משוכלים יותר ויתר. היום אנחנו משתמשים בהם בשבייל מושגים הדרושים יכולות reasoning משוכילות. לא כל מודל שפה אפילו הגודלים והחזקים יודעים לבצע reasoning המורכב מכמה שלבים בצורה חלקה. גם למידת few shot קלומר כשותנים למודל שפה מסויר מצומצם של שאלות הכלולות reasoning מתוכיהם והתשובה לא תמיד מספיק כדי "לכונן אותו" בצורה מספיק טוביה.

אוקי, אז נשמע שאנחנו בכיוון של לכיל מודל שפה על דאטאטס גדול ומוגון המכיל זוגות של שאלות ותשובות מנומקות עליהם מוסברות בשלבים (או בהרבה). דאטאטטים כאלה לא קיימים בכמות גדולה מספיק ויצירתם דורשת מאמצ גדול ויקר. אז המאמר המסורק מציע לבנות דאטאטס צזה מדאטאטיטים המכילים רק זוגות של שאלות ותשובות. בשבייל כך המחברים לקחו הרבה מאוד מדאטאטיטים מגוונים (כמה מאות אם לא אלפיים) וטיבו מודל שפה גדול (המלקחו את BLOOM) על דאטאטיטים אלו.

המ אימנו 3 מודלים נפרדים ממודל שפה הבסיסי זהה. הראשון הוא מיועד ללמידה עקרונית (learning fundamentals), השני למה שנקרא COT (Chain-Of-Thought), השלישי מתמחה בהוצאה טענות שגויות הנפוצות ביותר שעלוות להביא לתשובה לא נכונה בשאלת.

אחרי שיש לנו ביד 3 מודלים אלה ניתן לעשות אוגנטציה של מדאטאטיטים המכילים שאלות ותשובות בלבד ולהוסיף להם שלבי reasoning וגם הרשימה של טעויות נפוצות. אז כרגע ניתן לקחת מודל שפה יחסית קטן, להעשיר את הדאטאטס לכיוול שיש לנו ולטיב את מודל השפה הקטן עם הדאטאטס זה. למשימות מיוחדות ולא שגרתיות ניתן לכיל את 3 מודלי שפה (המורים) עליהם לשיפור בביבזועים.



## Review 178, Short: CAPSFUSION: Rethinking Image-Text Data at Scale

<https://huggingface.co/papers/2310.19019>

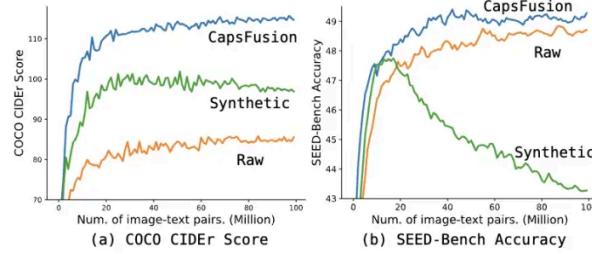


Figure 1. Training process of models trained on different captions.



Figure 2. (a) Comparison of raw and synthetic captions for training. (b) Data processing of Conceptual Captions [46], where real-world information is substituted with generic concepts .

הרבה דובר לאחרונה על איך ניתן לשפר מודלים מולטי-מודליים (אלו שידועים לעובוד עם כמה סוגים של נתונים, נגיד שפה ותמונות). רוב מודלים אלו אומנו על שדאותטיסים גדולים מהאינטרנט המכילים תמונות והכותרת שלהם. המודל הראשון המפורסם שאימן מודל זהה היה **CLIP**.

מכיוון שמודלים אלו מאמנים עם כמות עצומה (עשרות מיליוןים של תמונות או יותר) של נתונים בדיקה של איות הדטה אינה אפשרית. אך איך ניתן לשפר את איכות התאמה בין התמונה לכותרת שלא לשימוש בבודקים אנושיים. ניחשTEM נכון להשתמש ב-ChatGPT.

איך עושים זאת? לוקחים מודל שמאומן לתת כותרות לתמונות ויוצרים כותרת לתמונה איתו. לאחר מכן לוקחים את הכותרת המקורי ומבקשים מ-ChatGPT למצג אותן יחד לכותרת אחת עם איזה פרופט מהונדס היבט. ד"א הפרופט מכיל הוראה המונעת מ-ChatGPT מיזוג של שתי הכותרות. באמצעות פעולה פשוטה שצוו איכות הכותרת וישראל לתמונה משתפרת מאוד. בסוף לוקחים את הדטה עם הכותרות החדשות ומכילים על זה מודל מולטי-מודאי. לפי התוצאות במאמר עוזב לא רע.

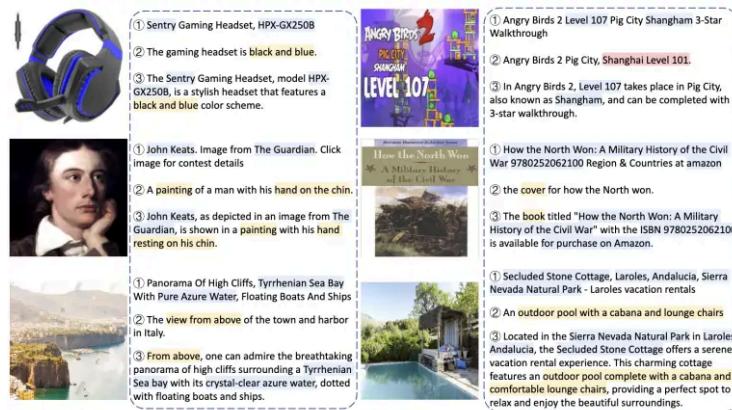


Figure 4. Examples of ① raw captions (from LAION-2B), ② synthetic captions (from LAION-COCO, generated by BLIP), and their corresponding ③ CAPSFUSION captions. Knowledge from raw captions (in blue) and information from synthetic captions (in yellow) are organically fused into integral CAPSFUSION captions. CAPSFUSION captions can also correct false information in synthetic captions (in red). More examples can be found in Fig. 8.

## Review 179, Short: UNLEASHING THE POWER OF PRE-TRAINED LANGUAGE MODELS FOR OFFLINE REINFORCEMENT LEARNING

<https://huggingface.co/papers/2310.20587>

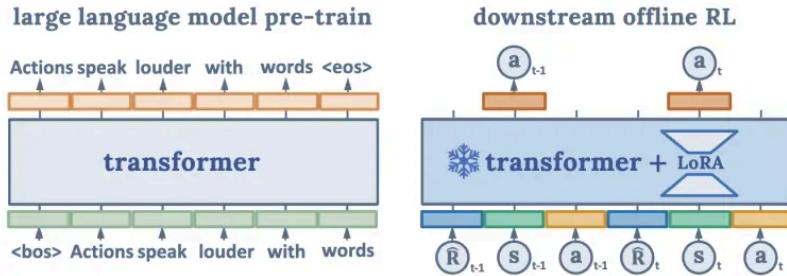


Figure 2: **The overview of LaMo.** LaMo mainly consists of two stages: (1) pre-training LMs on language tasks, (2) freezing the pre-trained attention layers, replacing linear projections with MLPs, and using LoRA to adapt to RL tasks. We also apply the language loss during the offline RL stage as a regularizer.

המאמר הזה משר את עיני כי אני מאוד אוהב שילובים של מודלי שפה (וגם מודלי דיפוזיה) למשימות מהעולם של לימודי עם חיזוקים (Reinforcement Learning). המאמר הזה עשה את זה בצורה מאוד אלגנטית כל כך אחרי שרפרפטי בו 5 דקות לא הבנתי חשבתי שהוא די בלתי אפשרי. אבל אחרי הקצת צלaltı לעומק הבנתי שכדי לתת לזה צ'אנס ולסקור אותו במדורנו.

از מה המאמר עשה בעצם? בגודל הם לקחו מודל שפה וטייבו (finetuned) אותו לבצע למשימות של RL. כמובן בהינתן של פעולות ומצבים קודמים המטרת של המודל היא לחזות את הפעולה הבאה. במקרה הזה מדובר באופליין RL כלומר המטרת של המודל היא לבדוק כמה שייתר טוב את הפעולות המוצלחות מהدادה הסט (בהינתן הפעולות והמצבים הקודמים). במשימות שנדרשו במאמר הפעולות מתוארות בצורה מילולית.

כבר מרים את מודלי השפה מתקרבים? אוקי, קודם המחברים לקחו מודל שפה מאומן (GPT2) וטייבו אותו על הדאטסהט הנקרא WiKiText. בשלב השני מוסיפים למודל שפה כמוה שכבות של MLP ומאמנים אותו על הדאטסה של המשימה (נגיד משחק אטاري) ובនוסף מכילים מודל שפה עם LoRA (זכרים מה זה?). תוך כדי התהילה הזה מוסיפים איבר רגולריזציה המכיל לוס על הדאטסה של WiKiText כנראה כדי לא למודל לא לשוכח את המוונות הקודמות שלו. וזה וזה מקבלים מודל-ל-RL כלומר decision transformer עם ביצועים טובים.

## Review 180, Short: Learning From Mistakes Makes LLM Better Reasoner

<https://huggingface.co/papers/2310.20689>

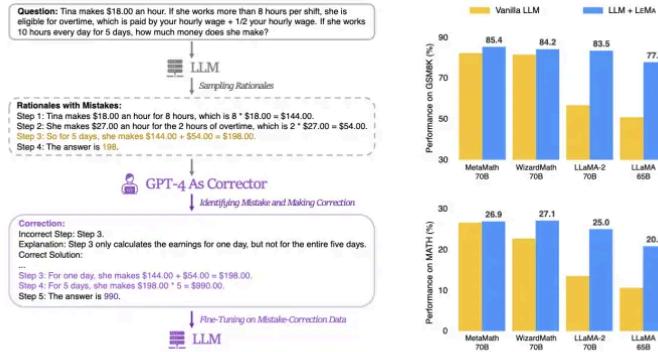


Figure 1: Left: The overall process of LEarning from MistAkes (LEMA). Right: The performance of LEMA on GSM8K (upper right) and MATH (lower right).

טוב, שוב בשבת אנחנו במאמר קليل וקצר על מודלי שפה גדולים. מתרברר שמודלי שפה בדומה לבני אדם מסוגלים ללמידה מההטיויות שלהם. כמובן אם אנו נכihil מודל שפה על הדатаה המכיל תיקונים של טעויות חסיבה המודל ילמד יותר טוב. כאמור המאמר לוקח מספר דוגמאות ומבקש ממודל שפה לבנות שרשרות חשיבה (Chain-of-Thought- CoT) עברון.

לאחר מכך ומפעלים על שרשרות חשיבה אלו מודל שפה חזק (כמו GPT4) וմבקשים ממנו לחפש שגיאות. המודל מתבקש לאתר שלבים המכילים שגיאות ולתקן אותן עד הפתרון. המחברים מצאו ש- GPT4 די טוב במשימה זו והצליח למצוא שלבים בעיתים ברוב שרשרות החשיבה.

از מה שהמחברים עשו הם לקחו נתונים של (שאלות, ותשובות), הפעילו את GPT4 עליו כדי לבנות שרשרת חשיבה. הם יצרו כמה שרשרות חשיבה לכל שאלה ופתרו שרשרות על תשבות לא נכונות. לאחר מכן הם ציינו מודל שפה(פחות חזק מ- GPT4) על הדטהה זהה. בשלב הבא הם הפעילו את GPT4 כדי לתקן שגיאות בשרשאות הלא נכונות וטיבבו (finetune) את המודל עליו. כתוצאה לכך הביצועים של המודל במשימה השתפה זו זה – קليل וקצר לשbeta.

**Example 1: Prompt For Generating Corrections**

For the following math word problems, the original solutions may contain errors. Please identify the incorrect step in each solution, explain why it is incorrect, and provide the correct solution starting from that step.

**Question:** James creates a media empire. He creates a movie for \$2000. Each DVD cost \$6 to make. He sells it for 2.5 times that much. He sells 500 movies a day for 5 days a week. How much profit does he make in 20 weeks?

**Original Solution:**

Step 1: 500 movies a day, 5 days a week, for 20 weeks, he sells  $500 * 5 * 20 = 50000$  movies.  
 Step 2: Each movie he sells for  $\$6 * 2.5 = \$15$ .  
 Step 3: Thus, he makes a profit of  $\$15 - \$6 = \$9$  per movie.  
 Step 4: Therefore, he makes a profit of  $\$9 * 50000 = \$450000$ .  
 Step 5: The answer is 450000.

**Incorrect Step:** Step 4.  
**Explanation:** The error in Step 4 is that it does not take into account the initial cost of creating the movie (\$2000). The correct solution should subtract this amount from the total profit calculated in Step 4.

**Correct Solution:**

...  
 Step 4: Therefore, he makes a profit of  $\$9 * 50000 - \$2000 = \$448000$ .  
 Step 5: The answer is 448000.

... (Another 3 examples)

**Question:**  $q_i$   
**Original Solution:**  $\tilde{r}_i$   
**Incorrect Step:** {Completion}