In-Context Ensemble Learning from Pseudo Labels Improves Video-Language Models in Low-Level Workflow Understanding

Moucheng Xu[1*†] **Evangelos Chatzaroulas**[1*] **Luc McCutcheon**[1] **Abdul Ahad**[1]

Hamzah Azeem[1] **Janusz Marecki**[1]

Ammar Anwar[1]
[1]Agile Loop, London, UK
moucheng.xu.18@alumni.ucl.ac.uk
Code: https://github.com/moucheng2017/SOP-LVM-ICL-Ensemble

Abstract

A Standard Operating Procedure (SOP) defines a step-by-step written guide for a business software workflow. SOP generation is a crucial step towards automating end-to-end software workflows. Manually creating SOPs can be time-consuming. Recent advancements in large video-language models offer the potential for automating SOP generation by analyzing recordings of human demonstrations. However, current large video-language models face challenges with zero-shot SOP generation. In this work, we first explore in-context learning with video-language models for SOP generation. We then propose In-Context Ensemble Learning, to aggregate pseudo labels of SOPs. The proposed in-context ensemble learning increases test-time compute and enables the models to learn beyond its context window limit with an implicit consistency regularisation. We report that in-context learning helps video-language models to generate more temporally accurate SOPs, and the proposed in-context ensemble learning can consistently enhance the capabilities of the video-language models in SOP generation.

1 Introduction

Video-language models are an emerging family of large foundational models attracting growing research interest. These models typically pre-train a visual encoder to project visual inputs into tokens, which are then utilised by large language models to interpret visual signals alongside text instructions. Recent video-language models have achieved significant success in high-level video understanding, such as high-level video summarisation, as reflected in their strong performance across diverse VQA benchmarks Maaz et al. [2024], Lin et al. [2024], Chen et al. [2024b]. However, current video-language models still face challenges when it comes to more complicated tasks. For instance, most available models have limited context windows for long videos or multiple short videos. Another challenge is their lack of competence in complex low-level video understandingWornow et al. [2024]. Standard Operating Procedure (SOP) generation is a typical low-level video understanding task Wornow et al. [2024]. SOP involves detailed, step-by-step descriptions of the chronological

[*] Equal contribution.

[†] Work completed at Agile Loop, Moucheng has now moved to Medtronic.

Workshop on Video-Language Models at 38th Conference on Neural Information Processing Systems (NeurIPS 2024).

ordering of actions taken in a software workflow on a recorded screen. The automatic generation of accurate SOPs can pave the way for the automation of business software.

In this work, we focus on improving the frontier video-language models in SOP generation from two perspectives: fine-tuning and increasing test-time compute. Since a lot of business software are domain specific, SOP generation is a novel task to the large models. One promising paradigm to fine-tune pre-trained large models to new unseen tasks is to use In-Context Learning Brown et al. [2020], Von Oswald et al. [2023], Pan et al. [2023], Akyürek et al. [2023], Bietti et al. [2023], Wei et al. [2022b,a]. In-Context Learning includes both the training samples and testing questions during the prompt. Unlike supervised fine-tuning, in-context learning requires no model parameter updates. In our case, we need to include training videos as a part of the multimodal prompt. However, in our preliminary exploratory experiments, we found that video-language models cannot receive as many video-training samples as the text- or images-training counterparts. This leads to an issue that the models hit the context window limit before it sees enough number of training videos, hampering the usability of in-context learning. Apart from fine-tuning, we also focus on extending the test-time computation. This is a common strategy for complicated reasoning tasks Wang et al. [2023], Wei et al. [2022c], Yao et al. [2023], Madaan et al. [2023], Snell et al. [2024], Khan et al. [2024] based on the intuition that humans normally spend more time generating multiple plausible reasoning paths before deciding the next action in complicated cognitive tasks. In order to increase the test-time compute without violating the context window limit at the same time, we present a new in-context learning strategy called In-Context Ensemble (ICE) Learning to achieve both the targets simultaneously. In summary, our main contributions of this paper can be summarized into two-fold:

1. We present the first evaluation of multimodal in-context learning in SOP generation with video-language models. We report that in-context learning consistently helps models to predict the step-by-step actions in a more accurate temporal order.
2. We introduce a multimodal in-context ensemble learning approach to increase the test-time compute of the video-language models in SOP generation with pseudo-labels. The in-context ensemble learning enables the models to learn from more video examples beyond their context window limit, with a self regularisation effect. We report that the proposed multimodal in-context ensemble learning consistently improves the models' predictions.

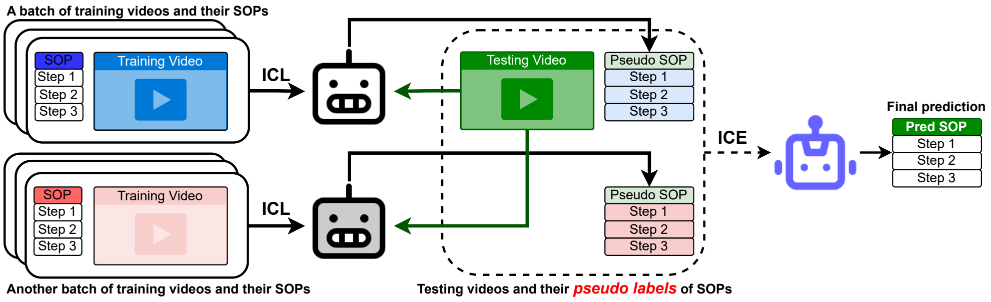


Figure 1: The proposed Multimodal In-Context Ensemble (ICE) learning with pseudo-labels. ICL: in-context learning. SOP: standard operating procedure, detailing chronological step-by-step actions in the video.

2 In-Context Ensemble Learning

By combining pseudo labelling Lee [2013], Xu et al. [2024, 2022b, 2024], Asano et al. [2020] with In-Context Learning (ICL) Brown et al. [2020], Von Oswald et al. [2023], Pan et al. [2023], Akyürek et al. [2023], Bietti et al. [2023], Wei et al. [2022b,a], we present Multimodal In-Context Ensemble (ICE) learning to enhance video-language models' understanding of low-level actions in videos. An illustration of the ICE pipeline is shown in above Fig 1. Our ICE pipeline first applies multimodal ICL to train several video-language models on different batches of training videos, along with their gold standard SOP text labels. Those fine-tuned video-language models then create different pseudo labels of SOPs from the testing videos. Finally, another video-language model processes the testing videos, along with those different pseudo labels as priors, to generate the final predictions of the SOPs. The proposed ICE pipeline offers four key benefits:

1. The introduction of the pseudo labelling in the reasoning process linearly scales up the test-time compute of the video-language models, improving the models' reasoning ability. Our ICE can be seen as an analogy to a black-box version of test-time tree search for aggregating multiple potential reasoning paths. But our ICE is much simpler than the test-time tree search methods.

2. Secondly, the ICE stage enables the model to learn from more examples beyond its context window limit. For instance, we could only feed up to 8 high-resolution videos as training examples to GPT-4o-mini due to the context window length. However, ICE allows the models to learn from the equivalent of 24-videos rather than 8-videos.

3. The third benefit is that the provided pseudo-labels sometimes share similar actions with variations. This implicit self consistency regularisation also aids the model's reasoning.

4. Lastly, the model does not need to utilise the entire context window to store the training video examples in the prompt, allowing it to focus more on the testing video. This is to address a common drawback of the large models that they struggles with information retrieval in extremely long tokens (e.g. needle search in a haystack) Kuratov et al. [2024], Lewis et al. [2020], Gu and Dao [2024], Ye et al. [2024].

3 Experiments

Data The WONDERBREAD benchmark Wornow et al. [2024] is a recently curated benchmark containing 2928 videos of distinct workflows and the detailed SOPs of the chronological step-by-step actions in the videos. We use a subset called "Gold Demo" from the WONDERBREAD benchmark of 724 videos. We randomly split the data into two parts, 30% training (217 videos) and 70% testing (507 videos). We subsequently split the training data into 28 batches, each batch contains 8 videos. Due to the limited computational resources, we only used the first 3 training batches.

Baseline 1: zero-shot We aim to evaluate the effectiveness of in-context learning with video-language models in SOP generation. The original zero-shot evaluation in the WONDERBREAD benchmark Wornow et al. [2024] injects additional information such as intent (a high-level summary of the video) and action trace (a detailed log of all user interactions with the screen, including elements' coordinates). We argue that providing such additional information to the models may not necessarily reflect the actual reasoning abilities of the foundational models. Therefore, we only feed the models the raw visual information, namely the key frames of the videos.

Baseline 2: ensemble We want to compare our multimodal in-context ensemble against traditional ensemble with video-language models in SOP generation. Therefore, our second baseline is a majority voting ensemble of the pseudo labels in text using the models, referred to as "Ensemble". We prompt the large models (GPT-4o-mini and Gemini-1.5-flash) to do automatic majority voting ensemble of the pseudo labels.

Video-Language Models As our approach requires long context windows with supports for multiple rounds of conversations, our choices of the models are limited to mostly closed models. Further considering the trade-off between performance and cost, we use GPT-4o-mini (OpenAI) and Gemini-1.5-flash (Google). We also include a zero-shot evaluation of a very recent public model from Microsoft, Phi-3.5. Throughout the experiments, to accommodate the context length window limits, if a video is longer than 20 frames, we down-sample the video to 20 frames. We also keep the videos at a high resolution of 1024 x 768.

Metrics Precision measures how many steps in the prediction match those in the gold standard SOP. Recall measures how many steps of the gold standard SOP are included in the predictions. Temporal order evaluates whether the sequence of steps in the prediction aligns with the sequence in the gold standard SOP. All evaluations are performed using GPT-4o-mini, following the original guidelines in Wornow et al. [2024].

Method	**Training Data**	**Precision (%) ↑**	**Recall (%) ↑**	**Temporal Order (%) ↑**
GPT-4o mini				
zero-shot	n/a	42.62	78.13	32.93
8-shot ICL	Batch 1	43.16	78.13	33.46
8-shot ICL	Batch 2	**45.95 (+3.33)**	78.13	34.15
8-shot ICL	Batch 3	44.51	78.13	33.08
Ensemble	Batch 1 - 3	36.05	72.31	21.47
ICE	Batch 1 - 3	44.34 (+1.72)	**84.79 (+6.66)**	**37.17 (+4.24)**
Gemini-1.5 flash				
zero-shot	n/a	34.35	45.16	27.82
8-shot ICL	Batch 1	34.10	45.18	35.15
8-shot ICL	Batch 2	33.99	41.96	29.42
8-shot ICL	Batch 3	34.08	40.75	29.77
24-shot ICL	Batch 1 - 3	29.75	39.42	26.47
Ensemble	Batch 1 - 3	30.30	40.52	26.12
ICE	Batch 1 - 3	**40.77 (+6.42)**	**54.38 (+9.22)**	**35.89 (+8.07)**
GPT-4o mini + Gemini-1.5 flash				
ICE	Batch 1 - 3	41.54	83.34	34.33
Phi-3.5				
zero-shot	n/a	31.66	45.88	24.42

Table 1: Testing results on 507 videos from the “Gold Demo” subset of WONDERBREAD benchmarkWornow et al. [2024]. Our evaluation is more challenging than the original one in Wornow et al. [2024] as we only feed videos into the models, *without* trace information (e.g. mouse clicks). ICL: in-context learning. ICE: in-context ensemble. Ensemble: majority voting of pseudo labels

Results As shown in Tab. 3, we observe that GPT-4o-mini generally outperforms Gemini-1.5-flash. And Gemini-1.5-flash slightly surpasses Phi-3.5. In-context learning (ICL) consistently improves the models’ ability to predict more correct order of steps compared to zero-shot baselines across different settings. ICE further enhances the models’ performance for both GPT4o-mini and Gemini-1.5-flash. Notably, for Gemini-1.5-flash, ICE resulted in a 9.22% improvement in recall. With GPT-4o-mini, ICE achieves high recall at 84.79% but low precision at 44.34%, suggesting that, while the model can successfully predict most of the steps, it may have added too many intermediate steps that were omitted in the ground truth. ICE also significantly outperforms the majority voting "Ensemble", highlighting the necessity of multimodal training samples in SOP generation. Further investigations are required to understand why "Ensemble" underperforms its individual components. Interestingly, the 24-shot ICL experiment with Gemini-1.5-flash shows that the model’s prediction accuracy varies inversely with the number of training samples. This may suggest that the foundational models naturally struggles to attend to too many high-resolution images at once. However, it is interesting to see that our ICE works well to address that issue. More results including analysis, discussion on related works, examples of successful and failed cases can be found in the Appendix ABC.

4 Future Works

We identify a few research directions: 1) scaling the proposed ICE pipeline; 2) standarlisation of SOP format, such as the length; 3) improving current AI-based evaluation of the temporal order of SOPs; 4) improving the effectiveness of video-language models at learning from long videos; 5) integration of other prompting techniques, such as self-reflection Shinn et al. [2023], Madaan et al. [2023].

5 Conclusion

We conducted an exploratory evaluation of in-context learning for SOP generation and proposed an In-Context Ensemble (ICE) learning approach using pseudo-labels for SOP generation. Our preliminary results suggest that the proposed ICE pipeline is effective in enhancing foundational video-language models in SOP generation. However, we also observe that the current performance of available foundational models has not yet reached a satisfactory level in SOP generation, highlighting the challenges of low-level video understanding from visual inputs alone.

References

Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. What learning algorithm is in-context learning? investigations with linear models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=0g0X4H8yN4I.

Yuki M. Asano, Christian Rupprecht, and Andrea Vedaldi. Self-labelling via simultaneous clustering and representation learning. In *International Conference on Learning Representations (ICLR)*, 2020.

Alberto Bietti, Vivien Cabannes, Diane Bouchacourt, Herve Jegou, and Leon Bottou. Birth of a transformer: A memory viewpoint. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 1560–1588. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/0561738a239a995c8cd2ef0e50cfa4fd-Paper-Conference.pdf.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

Guoxin Chen, Minpeng Liao, Chengxi Li, and Kai Fan. Alphamath almost zero: Process supervision without process, 2024a.

Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15750–15758, 2021.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, Bin Li, Ping Luo, Tong Lu, Yu Qiao, and Jifeng Dai. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 24185–24198, 2024b.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *CoRR*, abs/2110.14168, 2021. URL https://arxiv.org/abs/2110.14168.

Xidong Feng, Ziyu Wan, Muning Wen, Stephen Marcus McAleer, Ying Wen, Weinan Zhang, and Jun Wang. Alphazero-like tree-search can guide large language model decoding and training, 2023.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces, 2024. URL https://arxiv.org/abs/2312.00752.

Akbir Khan, John Hughes, Dan Valentine, Laura Ruis, Kshitij Sachan, Ansh Radhakrishnan, Edward Grefenstette, Samuel R. Bowman, Tim Rocktäschel, and Ethan Perez. Debating with more persuasive LLMs leads to more truthful answers. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 23662–23733. PMLR, 2024.

Yuri Kuratov, Aydar Bulatov, Petr Anokhin, Dmitry Sorokin, Artyom Sorokin, and Mikhail Burtsev. In search of needles in a 11m haystack: Recurrent memory finds what llms miss, 2024.

Dong-Hyun Lee. Pseudo-label : The simple and efficient semi-supervised learning method for deep neural networks. 2013. URL https://api.semanticscholar.org/CorpusID:18507866.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/6b493230205f780e1bc26945df7481e5-Paper.pdf.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=v8L0pN6EOi.

Ji Lin, Hongxu Yin, Wei Ping, Pavlo Molchanov, Mohammad Shoeybi, and Song Han. Vila: On pre-training for visual language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 26689–26699, 2024.

Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Khan. Video-ChatGPT: Towards detailed video understanding via large vision and language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12585–12602, Bangkok, Thailand, August 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.acl-long.679.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-refine: Iterative refinement with self-feedback. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 46534–46594. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/91edff07232fb1b55a505a9e9f6c0ff3-Paper-Conference.pdf.

Jane Pan, Tianyu Gao, Howard Chen, and Danqi Chen. What in-context learning "learns" in-context: Disentangling task recognition and task learning. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8298–8319, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.527. URL https://aclanthology.org/2023.findings-acl.527.

Shuvendu Roy and Ali Etemad. Consistency-guided prompt learning for vision-language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=wsRXwlwx4w.

William Saunders, Catherine Yeh, Jeff Wu, Steven Bills, Long Ouyang, Jonathan Ward, and Jan Leike. Self-critiquing models for assisting human evaluators, 2022.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: language agents with verbal reinforcement learning. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 8634–8652. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/1b44b878bb782e6954cd888628510e90-Paper-Conference.pdf.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, Timothy P. Lillicrap, Karen Simonyan, and Demis Hassabis. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *CoRR*, abs/1712.01815, 2017. URL http://arxiv.org/abs/1712.01815.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling model parameters, 2024.

Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D. Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Neural Information Processing Systems (NeurIPS)*, 2020.

Johannes Von Oswald, Eyvind Niklasson, Ettore Randazzo, Joao Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. Transformers learn in-context by gradient descent. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 35151–35174. PMLR, 2023. URL `https://proceedings.mlr.press/v202/von-oswald23a.html`.

Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. Math-shepherd: Verify and reinforce LLMs step-by-step without human annotations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.510. URL `https://aclanthology.org/2024.acl-long.510`.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=1PL1NIMMrw`.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*, 2022a. URL `https://openreview.net/forum?id=gEZrGCozdqR`.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*, 2022b.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language models. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022c. URL `https://openreview.net/forum?id=_VjQlMeSB_J`.

Michael Wornow, Avanika Narayan, Ben Viggiano, Ishan S. Khare, Tathagat Verma, Tibor Thompson, Miguel Angel Fuentes Hernandez, Sudharsan Sundar, Chloe Trujillo, Krrish Chawla, Rongfei Lu, Justin Shen, Divya Nagaraj, Joshua Martinez, Vardhan Agrawal, Althea Hudson, Nigam H. Shah, and Christopher Re. Do multimodal foundation models understand enterprise workflows? a benchmark for business process management tasks, 2024.

Mou-Cheng Xu, Yukun Zhou, Chen Jin, Stefano B. Blumberg, Frederick J. Wilson, Marius de Groot, Daniel C. Alexander, Neil P. Oxtoby, and Joseph Jacob. Learning morphological feature perturbations for calibrated semi-supervised segmentation. *International Conference on Medical Imaging with Deep Learning (MIDL)*, 2022a.

Mou-Cheng Xu, Yukun Zhou, Chen Jin, Marius de Groot, Daniel C. Alexander, Neil P. Oxtoby, Yipeng Hu, and Joseph Jacob. Bayesian pseudo labels: Expectation maximization for robust and efficient semi-supervised segmentation. In Linwei Wang, Qi Dou, P. Thomas Fletcher, Stefanie Speidel, and Shuo Li, editors, *Medical Image Computing and Computer Assisted Intervention – MICCAI 2022*, pages 580–590, Cham, 2022b. Springer Nature Switzerland.

Mou-Cheng Xu, Yukun Zhou, Chen Jin, Marius de Groot, Daniel C. Alexander, Neil P. Oxtoby, and Joseph Jacob. Mismatch: Calibrated segmentation via consistency on differential morphological feature perturbations with limited labels. *IEEE Transactions on Medical Imaging (TMI)*, 2023.

Moucheng Xu, Yukun Zhou, Chen Jin, Marius de Groot, Daniel C. Alexander, Neil P. Oxtoby, Yipeng Hu, and Joseph Jacob. Expectation maximisation pseudo labels. *Medical Image Analysis*, 94:103125, 2024. ISSN 1361-8415. doi: https://doi.org/10.1016/j.media.2024.103125. URL `https://www.sciencedirect.com/science/article/pii/S1361841524000501`.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik R Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=5Xc1ecxO1h.

Tianzhu Ye, Li Dong, Yuqing Xia, Yutao Sun, Yi Zhu, Gao Huang, and Furu Wei. Differential transformer, 2024.

A Related Work

Ensemble Ensemble scales the inference compute by treating each candidate prediction equally to improve the base model's reasoning to reach a more accurate final prediction. The most common strategy of ensemble is majority voting. Majority voting picks up the most frequent prediction of all of the candidate predictions as the final prediction Wang et al. [2023], Chen et al. [2024a], Snell et al. [2024]. A lot of existing large models (e.g. Google Gemini, OpenAI GPT4) provide options for enabling majority voting during inference. We also use the majority voting ensemble as a baseline in our experiments. Different from traditional majority voting ensemble, our proposed ICE has a proposal sampling distribution with large variances. This is because the traditional majority voting ensemble samples candidate predictions from the same model Snell et al. [2024], whereas our ICE samples per candidate prediction from each differently fine-tuned model.

Consistency Prompts Consistency regularisation has been playing a pivotal role in semi-supervised and self-supervised learning Chen and He [2021], Xu et al. [2022a], Sohn et al. [2020], Xu et al. [2023]. The general idea is to apply a consistency restraint on the predictions from different experts that are fed with the same source input. A recent work Roy and Etemad [2024] also discovered that consistency regularisation improves the video-language models at few-shot learning while the inputs prompts are differently perturbed. Although our In-Context Ensemble (ICE) does not specifically implement the consistency regularisation. By learning from different pseudo labels, our ICE implicitly enforces consistency regularisation, because sometimes, the same actions do appear among different pseudo labels but with variations.

Test-Time Tree Search Tree search has been a popular tool to scale up test-time compute by aggregating the candidate sequential reasoning processes Chen et al. [2024a], Snell et al. [2024], Feng et al. [2023], Silver et al. [2017] for better complex reasoning. From the end-to-end perspective, our ICE shares similiarities with Test-Time Tree Search, especially Lookahead Search. Each SOP can be seen as a sequential reasoning process, changing the status of the screen from the first frame to the last frame, following a particular temporal path. Multiple SOP pseudo labels resemble a tree structure and the generation of the final SOP is a black-box version of steps aggregation of pseudo labels, thereby an analogy to Tree Search methods. However, there are noticeable differences between the Tree Search methods and our ICE. In Beam Search or Lookahead Search, a Verifier Lightman et al. [2024], Wang et al. [2024], Cobbe et al. [2021] is used to select the action step-wise, at each level of the tree, focusing on exploiting the options. Whereas in our ICE, we use a large video-language model to consider the pseudo labels to generate the final chain-of-the-steps, focusing on conditioning exploration, which is beyond exploiting available candidate pseudo labels. In addition, we do not train an extra Verifier and we do not have an explicit selection process per action.

Reflection Prompts Self-revision has also been another popular tool to scale up test-time compute by iteratively prompting the same large language model to refine its initial prediction Madaan et al. [2023], Saunders et al. [2022], Khan et al. [2024]. This approach is simple yet effective, computationally affordable too. In our ICE method, the models are implicitly asked to do self-reflection on the old predictions, which are the pseudo labels. However, different from the single model based reflection Madaan et al. [2023], Saunders et al. [2022], our ICE involved with multiple models which are respectively fine-tuned on different data. We argue that by doing so, we can increase the variance of the proposal distribution for better reasoning outcomes.

B Analysis

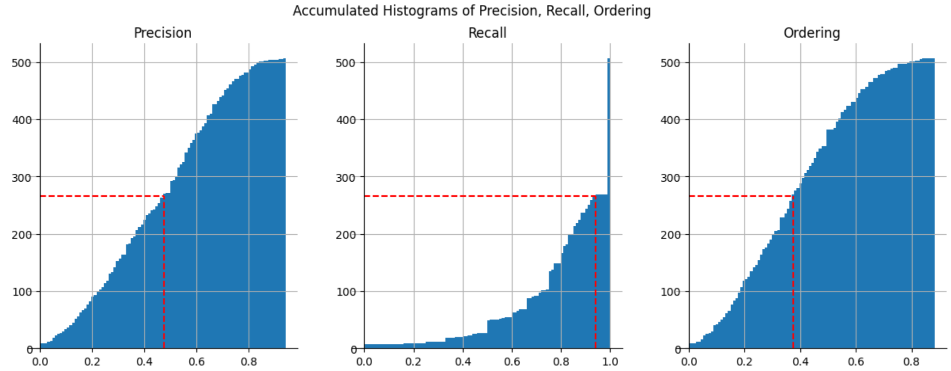


Figure 2: Accumulated histograms of testing results with the ICE with GPT-4o-mini. The red vertical lines indicate the 50-percentile of testing cases.

We visualised the accumulated histograms of the metrics in the above Fig.2. The 50th percentile of recall is above 90%, showing that the model is capable at predicting the actions in the ground truth. However, the 50th percentile of precision is lower than 50%, and the 50th percentile of temporal ordering is even lower at, less than 40%. We hypothesise that the unsatisfactory performance in precision and ordering metrics might be due to the model having its own preferences for the style in which the output steps are written. To explore this further, we examined the relationship between the lengths of the SOPs and the model's performance, as SOP length reflects the model's preferred writing style.

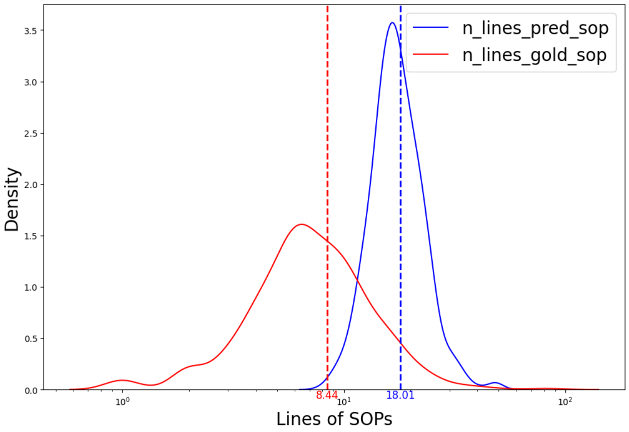


Figure 3: Kernel density estimate plots of lines of SOPs. ICE with GPT-4o-mini.

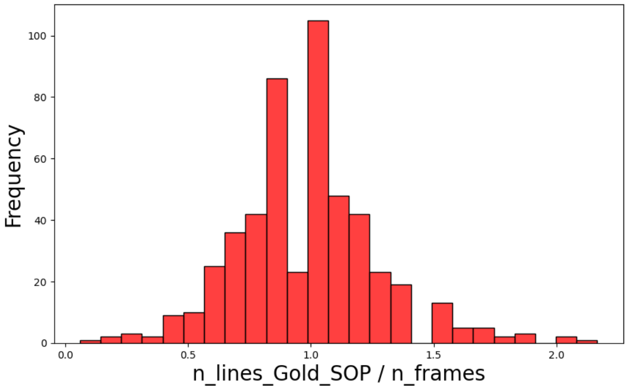


Figure 4: Histogram of ratios between number of lines of ground truth SOPs and the number of frames for each testing case.

In Fig.3, we plotted the kernel density estimates of both the gold and predicted SOPs. The predicted SOPs have an average of 18.01 steps, with most concentrated around this value. In contrast, the gold SOPs average 8.44 steps, with a more even distribution across cases. This suggests that the model fails to capture the diversity seen in the gold SOPs, implying that it may have developed its own writing style to write the SOPs, that is different from the styles shown in the training examples. Additionally, we plotted the ratios between the number of SOP lines and the number of frames in Fig.4, observing a wide distribution. This raises the possibility that even among the ground truth SOPs, there may be inconsistencies in writing styles, hindering the effectiveness of in-context learning. We then further grouped the predictions according to the length of their ground truth SOPs in the following Fig. 5, 6, 7. It is clear to see that, the more different between the lengths of the ground truth SOPs and the mean length of the predicted SOPs, the worse the performances.

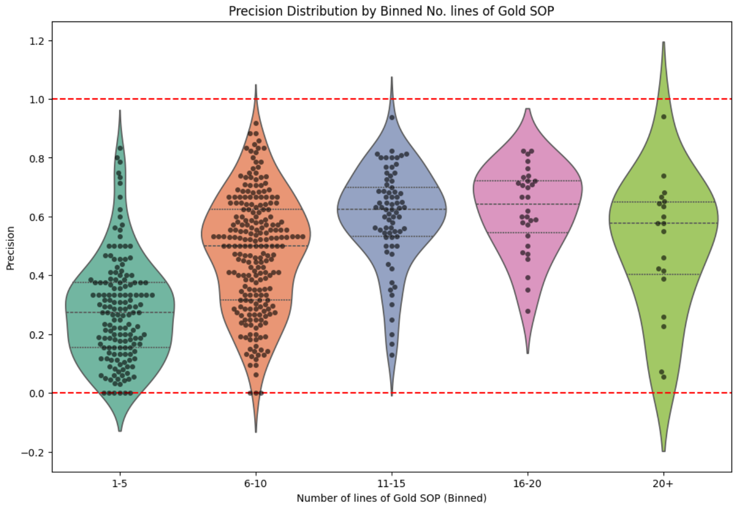


Figure 5: Violin plots of binned precision according to gold SOP lengths.

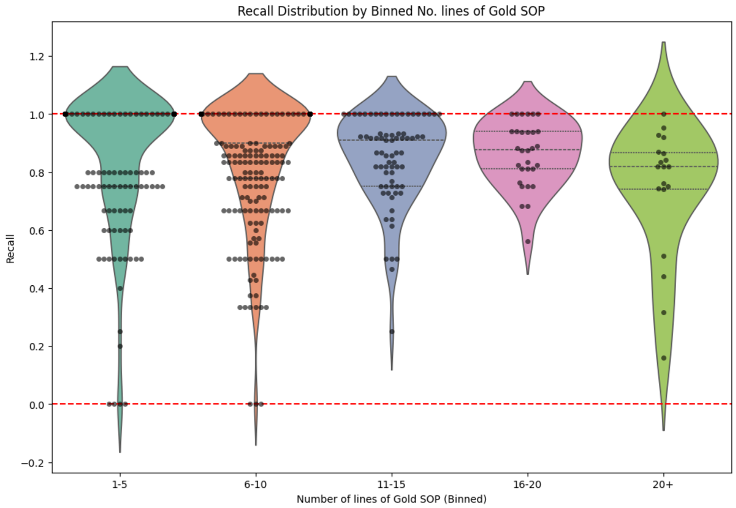


Figure 6: Violin plots of binned recall according to gold SOP lengths.

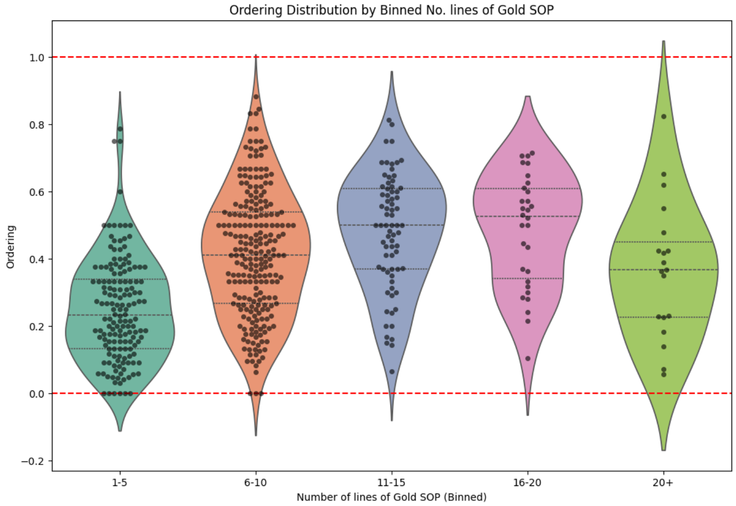
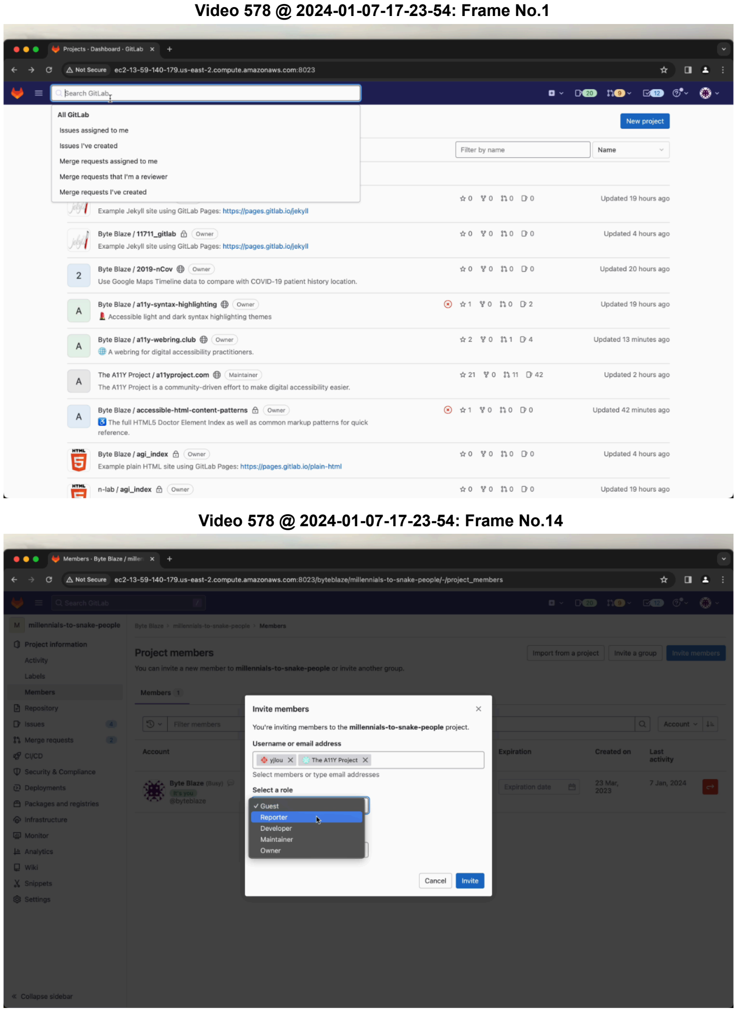


Figure 7: Violin plots of binned temporal ordering according to gold SOP lengths.

Based on our above observations, we propose following future research directions:

1. Investigation into standardised writing style of SOPs is necessary. For example, one step of the SOP for each frame, with a fixed number of frames per video.
2. Examination into the impact of the format of SOPs during prompt engineering might be beneficial.
3. The current GPT-4o-mini based evaluation of temporal ordering is limited; a better metric or evaluation pipeline is required to further our insight into the impact of the models.
4. Integration of other prompting techniques can further enhance the pipeline, such as self-reflection Shinn et al. [2023], Madaan et al. [2023].

C A Successful Case



Video 578 @ 2024-01-07-17-23-54: Frame No.1

Projects · Dashboard · GitLab

Not Secure ec2-13-59-140-179.us-east-2.compute.amazonaws.com:8023

Search GitLab

Example Jekyll site using GitLab Pages: https://pages.gitlab.io/jekyll — Updated 19 hours ago

Byte Blaze / 11711_gitlab (Owner) — Example Jekyll site using GitLab Pages: https://pages.gitlab.io/jekyll — Updated 4 hours ago

Byte Blaze / 2019-nCov (Owner) — Use Google Maps Timeline data to compare with COVID-19 patient history location. — Updated 20 hours ago

Byte Blaze / a11y-syntax-highlighting (Owner) — Accessible light and dark syntax highlighting themes — Updated 19 hours ago

Byte Blaze / a11y-webring.club (Owner) — A webring for digital accessibility practitioners. — Updated 13 minutes ago

The A11Y Project / a11yproject.com (Maintainer) — The A11Y Project is a community-driven effort to make digital accessibility easier. — Updated 2 hours ago

Byte Blaze / accessible-html-content-patterns (Owner) — The full HTML5 Doctor Element Index as well as common markup patterns for quick reference. — Updated 42 minutes ago

Byte Blaze / agi_index (Owner) — Example plain HTML site using GitLab Pages: https://pages.gitlab.io/plain-html — Updated 4 hours ago

n-lab / agi_index (Owner) — Updated 19 hours ago

Video 578 @ 2024-01-07-17-23-54: Frame No.14

Members · Byte Blaze / millen

Not Secure ec2-13-59-140-179.us-east-2.compute.amazonaws.com:8023/byteblaze/millennials-to-snake-people/-/project_members

Project members

You can invite a new member to **millennials-to-snake-people** or invite another group.

Invite members

You're inviting members to the **millennials-to-snake-people** project.

Username or email address: yjlou ✕ The A11Y Project ✕

Select members or type email addresses

Select a role: ✓ Guest, Reporter, Developer, Maintainer, Owner

Byte Blaze (Busy) It's you @byteblaze — Expiration date — Created on 23 Mar, 2023 — Last activity 7 Jan, 2024

Figure 8: Testing video (578-2024-01-07-17-23-54). Row 1: 1st frame from the video. Row 2: 9th frame from the video.

The task "578-2024-01-07-17-23-54" in Fig. 8 involves adding new users to a GitLab repository as reporters. The model used was ICE with GPT-4o-mini. As shown in Fig. 9, all steps of the ground truth were correctly predicted, but the prediction included extra details, highlighted in red. The metrics calculated by GPT-4o-mini for this task are: precision 81.25%, recall 93.75%, and temporal order correctness 68.75%. We observed that both workflows actually follow the same temporal order, though they are not perfectly synchronised. This highlights a limitation in the use of GPT-4o-mini for evaluation, suggesting that the model's foundational ability may lack some basic temporal understanding. Consequently, the model's performance may not be as poor as the metrics indicate.

Ground Truth

1. Click the purple/gray search bar on the top left of the screen (icon with three lines stacked).
2. Type ‘millennials-to-snake-people’.
3. Press enter.
4. Click the first pane on the top of the field of panes of projects that show up in the middle, with black text that has words like timeline, item, management, extension.
5. Click the black bold “Project Information” tab on the top left side of the screen above “Repository.”
6. Click the black bold “Members” text on the left side of the screen under “Labels.”
7. Click the blue “Invite members” button on the top right side of the screen.
8. Type "yjlou."
9. Click the pane with the colored avatar that pops up below.
10. Click the right side of the text box that says “Username or email address.”
11. Type "a11yproject."
12. Click the pane with the colored avatar that pops up below.
13. Click the right side of the text box that says “Username or email address.”
14. Click the dropdown underneath the black text saying “Select a role.”
15. Click “Reporter” from the dropdown.
16. Click the blue “Invite” button on the bottom right.

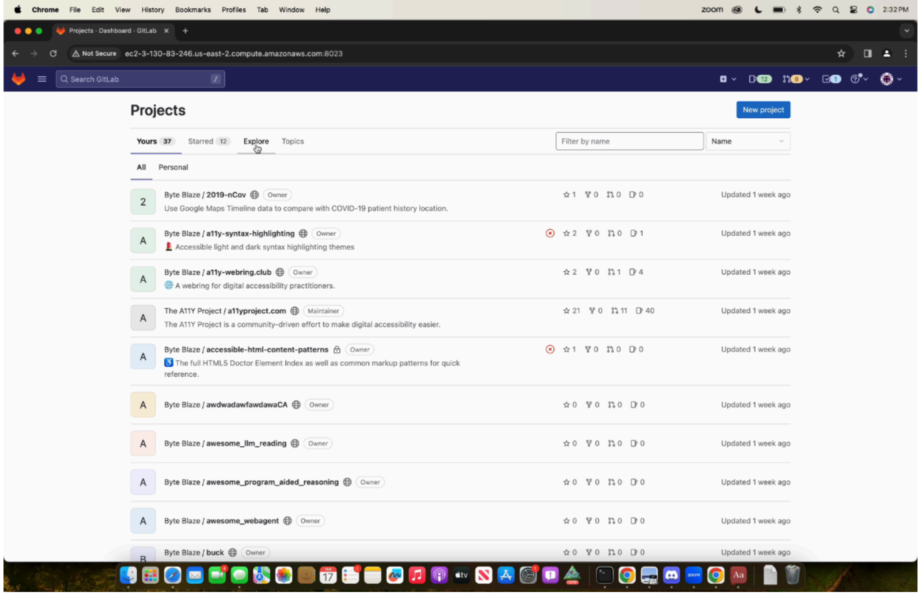
Prediction

1. Navigate to Projects: Click on the "Projects" tab located in the top navigation bar of the GitLab dashboard, positioned next to the "Dashboard" tab.
2. Search for Project: In the search bar at the top of the Projects page, which is labeled "Search GitLab," click to focus on it.
3. Type Project Name: Type "millennials-to-snake-people" into the search bar.
4. Select Project: Wait for the suggestions to load and then click on the project name "millennials-to-snake-people" from the dropdown suggestions to navigate to the project’s main page.
5. Access Members Section: In the left sidebar of the project page, click on "Project information" to expand the section.
6. Click Members: Under the expanded "Project information" section, click on the "Members" option to go to the members page.
7. Open Invite Members Dialog: In the Members section, locate and click the blue "Invite members" button on the top right side of the screen to open the invite members dialog.
8. Focus on Username Field: In the "Invite members" pop-up window, click into the text box labeled "Username or email address."
9. Type Username: Type "yjlou" into the text box to search for the user you want to invite.
10. Select User from Results: Wait for the system to display the search results. When "yjlou" appears in the results, click on it to select that user.
11. Open Role Dropdown: Below the selected username, click on the dropdown menu labeled "Select a role" to choose the role for the invited member.
12. Choose Role: From the dropdown options, select "Reporter" by clicking on it to set the user’s access level.
13. Leave Expiration Date Blank: If necessary, locate the "Access expiration date (optional)" field and leave it blank.
14. Send Invitation: Click the blue "Invite" button located at the bottom right of the pop-up window to send the invitation.
15. Confirm Invitation Success: After the invitation is sent, observe the confirmation message that appears, indicating that the member has been successfully added.
16. Review Members List: Finally, review the list of project members to ensure that "yjlou" now appears in the members’ list, confirming the successful invitation.

Figure 9: Comparison of the Ground Truth with the Prediction of "578-2024-01-07-17-23-54". Precision: 81.25%. Recall: 93.75%. Temporal order: 68.75%.

D A Failed Case

Video 317 @ 2024-02-17-14-32-37: Frame No.1



ec2-3-130-83-246.us-east-2.compute.amazonaws.com:8023

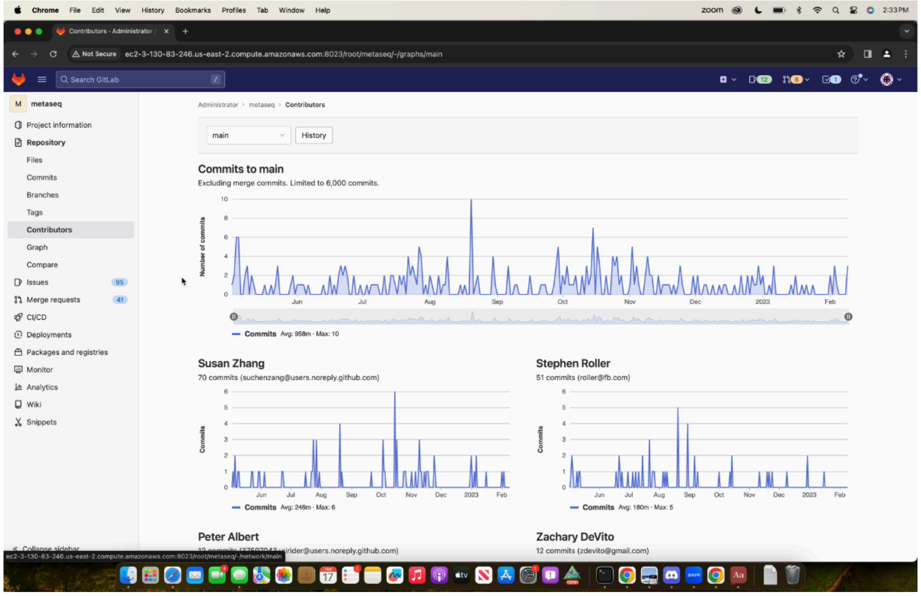
Projects

Yours 37 Starred 12

All Personal

Project	Stars	Forks	Merge requests	Issues	Updated
2 Byte Blaze / 2019-nCov (Owner) Use Google Maps Timeline data to compare with COVID-19 patient history location.	1	0	0	0	Updated 1 week ago
A Byte Blaze / a11y-syntax-highlighting (Owner) Accessible light and dark syntax highlighting themes	2	0	0	1	Updated 1 week ago
A Byte Blaze / a11y-webring.club (Owner) A webring for digital accessibility practitioners.	2	0	1	4	Updated 1 week ago
A The A11Y Project / a11yproject.com (Maintainer) The A11Y Project is a community-driven effort to make digital accessibility easier.	21	0	11	40	Updated 1 week ago
A Byte Blaze / accessible-html-content-patterns (Owner) The full HTML5 Doctor Element Index as well as common markup patterns for quick reference.	1	0	0	0	Updated 1 week ago
A Byte Blaze / awdwadawfawdawaCA (Owner)	0	0	0	0	Updated 1 week ago
A Byte Blaze / awesome_llm_reading (Owner)	0	0	0	0	Updated 1 week ago
A Byte Blaze / awesome_program_aided_reasoning (Owner)	0	0	0	0	Updated 1 week ago
A Byte Blaze / awesome_webagent (Owner)	0	0	0	0	Updated 1 week ago
B Byte Blaze / buck (Owner)	0	0	0	0	Updated 1 week ago

Video 317 @ 2024-02-17-14-32-37: Frame No.8



ec2-3-130-83-246.us-east-2.compute.amazonaws.com:8023/root/metaseq/-/graphs/main

Administrator › metaseq › Contributors

Commits to main

Excluding merge commits. Limited to 6,000 commits.

Commits Avg: 958m · Max: 10

Susan Zhang

70 commits (suchenzang@users.noreply.github.com)

Commits Avg: 246m · Max: 6

Stephen Roller

51 commits (roller@fb.com)

Commits Avg: 180m · Max: 5

Peter Albert

12 commits (37597042+...rider@users.noreply.github.com)

Zachary DeVito

12 commits (zdevito@gmail.com)

ec2-3-130-83-246.us-east-2.compute.amazonaws.com:8023/root/metaseq/-/network/main

Figure 10: Testing video (317-2024-02-17-14-32-37). This task involves listing the names and number of commits of the top three contributors to a GitLab repository called "metaseq." Row 1: 1st frame from the video. Row 2: 8th last frame from the video.

Ground Truth

1. Click on "Explore" under the "Projects" heading
2. Click on the search bar in the top right
3. Type in "metaseq"
4. From the results below, click on "Administrator / metaseq"
5. Click on the "Repository" tab in the left sidebar.
6. Click on the "Contributors" tab in the left sidebar.
7. Report the information of the top three people listed (top left, top right, second left)

Prediction

1. On the GitLab dashboard, locate the "Projects" tab in the top navigation bar, situated between the "Dashboard" and "Groups" options, and click on it.
2. In the "Projects" section, observe the list of projects displayed. Scroll down until you can see the project titled "The A11Y Project / a11yproject.com", which is the fourth project from the top.
3. Click on the project titled "The A11Y Project / a11yproject.com" to navigate to its main page.
4. Once on the project page, locate the "Project Information" tab on the top left side, directly under the project name, and click on it.
5. In the expanded options under "Project Information," click on the "Members" tab, located directly beneath the "Project Information" heading.
6. Click the blue “Invite members” button located at the top right of the members section, next to the “Invite a group” option.
7. In the popup window that appears, click on the text box labeled "Username or email address."
8. Type "Koushik" into the "Username or email address" text box.
9. Press Enter to select "Koushik Dutra" from the dropdown that appears as you type.
10. Click on the dropdown labeled "Select a role" located directly beneath the username text box.
11. Choose "Guest" from the options available in the dropdown list.
12. Click the blue "Invite" button located at the bottom right of the popup window inside the shaded area to send the invitation.
13. Confirm that the member has been successfully added by checking for a notification on the screen that states "Members were successfully added."

Figure 11: Comparison of the Ground Truth with the Prediction of "317-2024-02-17-14-32-37". Precision: 0%. Recall: 0%. Temporal order: 0%. The model’s prediction is a completely different SOP for adding new guests to the GitLab repository. This indicates that the model does not fundamentally understand the software.