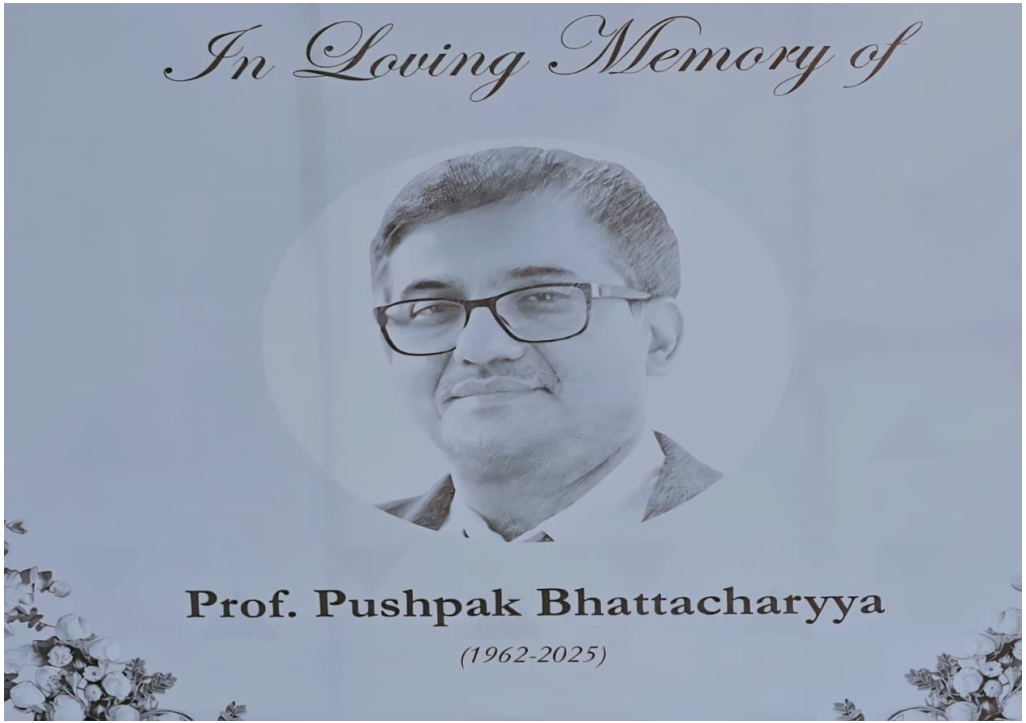
Continual Learning for Large Language Models

Dr. Srijith P. K.
Bayesian Reasoning and Intelligence(BRAIN) Lab
Department of Computer Science and Engineering
Department of Artificial Intelligence (Affiliated)
Indian Institute of Technology Hyderabad
https://www.iith.ac.in/cse/srijith/
https://sites.google.com/view/brainiith/home
srijith@cse.iith.ac.in

भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

Bayesian Reasoning And INtelligence



Thy will be done

Large Language Models (LLMs)

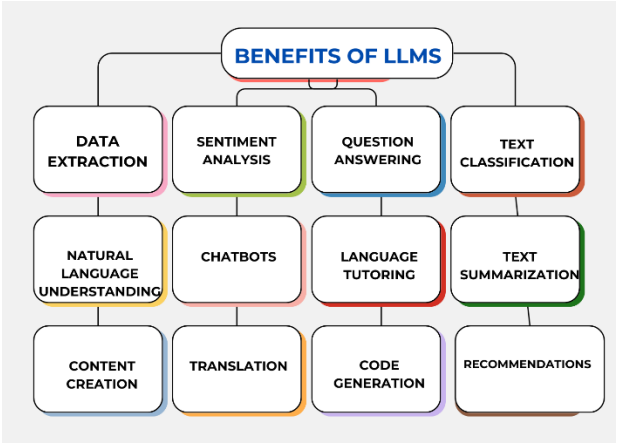
LLMs are AI models trained on vast text datasets to understand and generate human language, analyze data, automate tasks, and provide personalized insights.

Significance of LLMs in AI

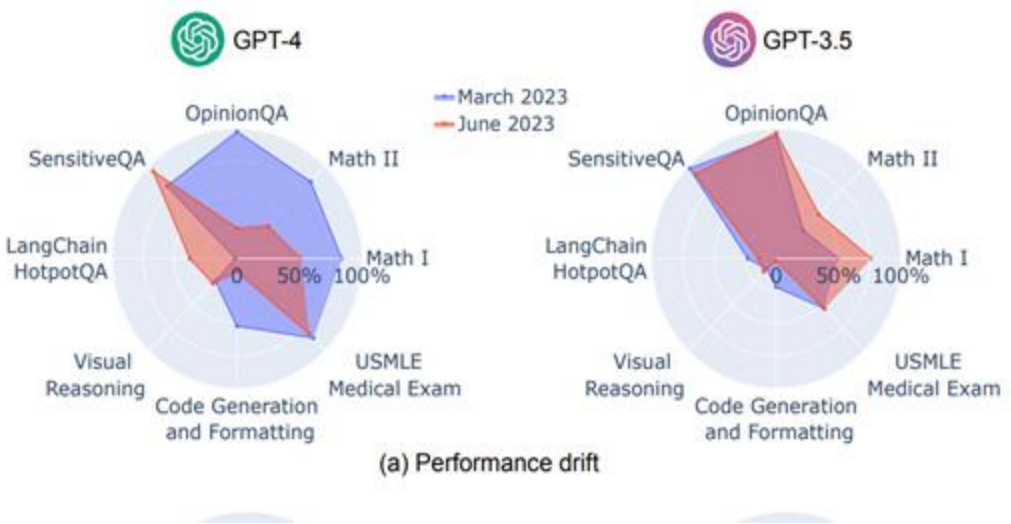
- **Enhanced Human-Computer Interaction**: LLMs enable dynamic, context-specific, and personalized AI, improving user experience.
- **Transformative Role in AI**: LLMs generate human-like text, enhancing creativity and personalization across industries like healthcare, content creation, and customer service

LLM Applications

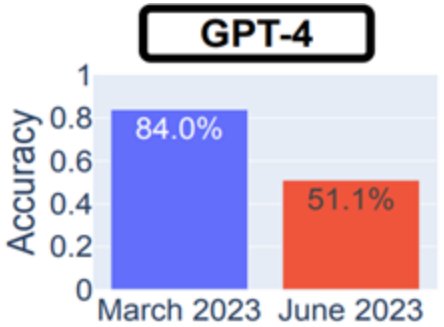
- **Customer Service**: AI chatbots powered by LLMs provide real-time, contextual responses, improving customer satisfaction.
- **Content Creation**: LLMs automate content generation, helping businesses scale marketing while maintaining quality.



Catastrophic Forgetting in LLMs



(a) Performance drift



Performance drop in Prime vs. Composite Classification of GPT4 model

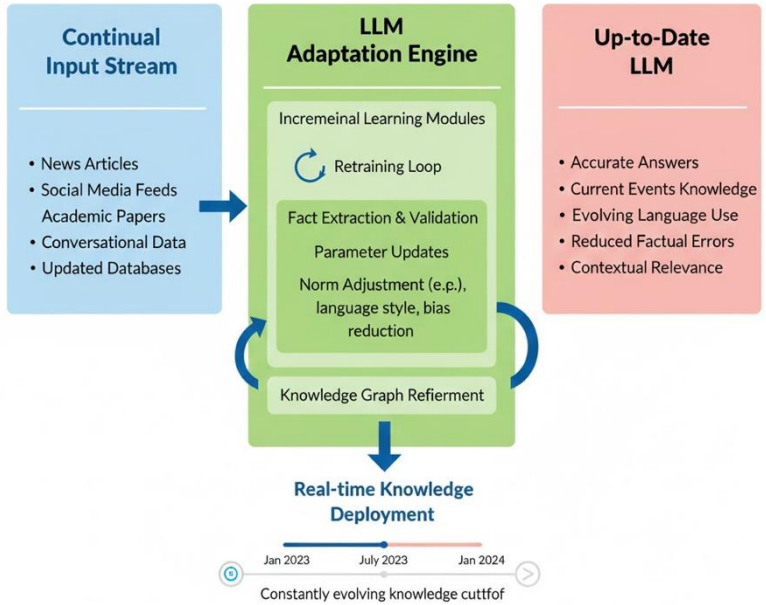
GPT-4 performance dropped from March to June 2023 across multiple categories (Math, Reasoning, OpinionQA, Code), while **GPT-3.5 improved** in several of the same tasks.

These opposing shifts show that model updates can **alter capabilities unpredictably**, reinforcing the need for more stable and continual learning approaches.

Chen, L., Zaharia, M. and Zou, J., 2024. How is ChatGPT's behavior changing over time?. *Harvard Data Science Review*, *6*(2).

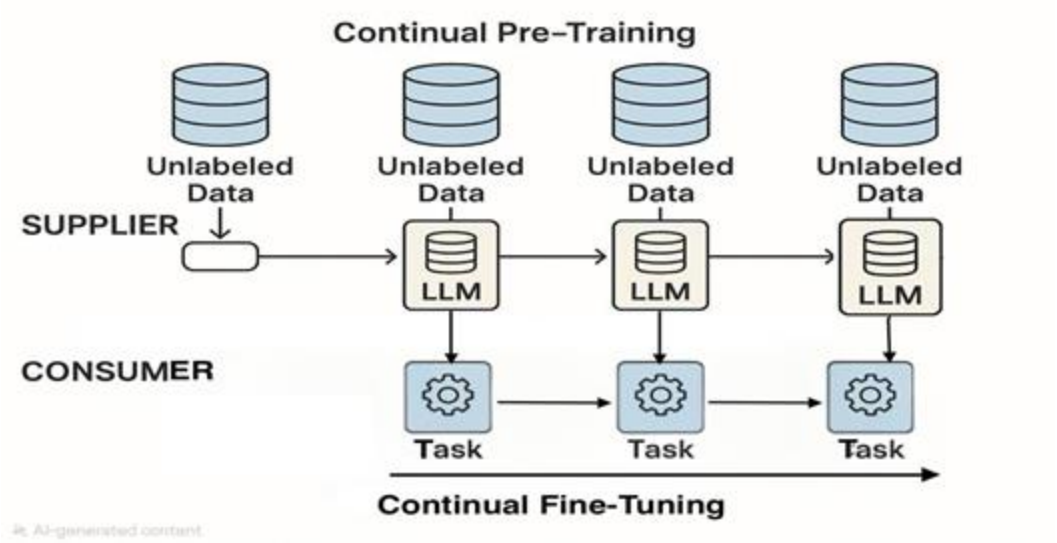
Continual Learning in LLMs

- **The world changes continuously**: facts, language, code, and norms evolve.
- LLMs are trained on **static data**, but needs to be adapted to the evolving world
- Retraining is **slow and expensive**
  - *Huge clusters of GPUs/TPUs over weeks or months.*
  - Thousands to millions of GPU hours
- **Continual learning** enables LLMs to update with new information, adapt to new domains and learn behaviours incrementally, while retaining the old knowledge.



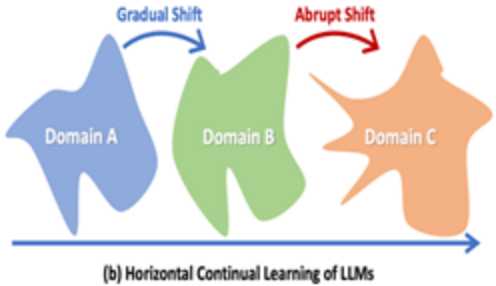
Large Language Models in Practice

Supplier consumer Model : On the supplier side, the model is pre-trained over a sequence of large-scale unlabeled datasets. After every release of the pre-trained model, the consumer utilizes the stronger and more up-to-date upstream model for downstream tasks. This cycle is repeated multiple times..



Continual Learning in LLMs

1. LLMs would require broadly 2 kinds of continual learning
2. **Horizontal continuity** lies in the dynamic nature of data distribution over time.
3. **Vertical continuity** is characterized by a hierarchical structure, the data distribution of upstream tasks partially covers the downstream



(b) Horizontal Continual Learning of LLMs

Horizontal Continuity the model needs to handle **overlapping representations** and **update** and **addition** of new data and preferences.

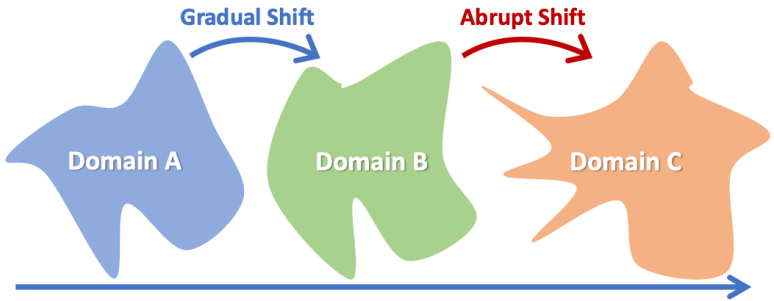
(a) Vertical Continual Learning of LLMs

Vertical tasks often **reuse** the **same input-output mapping**; less parameter interference. Parameter-efficient methods (LoRA, adapters) let us isolate minute adjustments cheaply.

Continual Learning of Large Language Models: A Comprehensive Survey. ACM Comput. Surv. 58, 5, Article 120 (April 2026)

CL in LLM : Horizontal Continuity

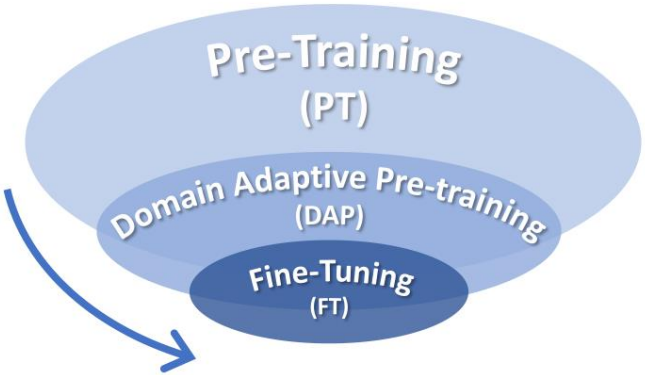
- Horizontal continuity lies in the dynamic nature of data distribution over time.
- LLMs struggle to retain complete knowledge of past experiences when adapting to new temporal domains, although they do demonstrate a higher level of robustness against catastrophic forgetting
- **Long Task Sequences**. Horizontal continual learning ideally involves numerous incremental phases
- **Abrupt Distributional Shift. I**n contrast to vertical continuity, here distributional shifts are abrupt and unpredictable



(b) Horizontal Continual Learning of LLMs

CL in LLM : Vertical Continuity

- Vertical continuity is characterized by a hierarchical structure encompassing data inclusiveness, and computational resource
- The data distribution of upstream tasks partially covers the downstream : model might start of at a decent initialization for the subsequent stage of training

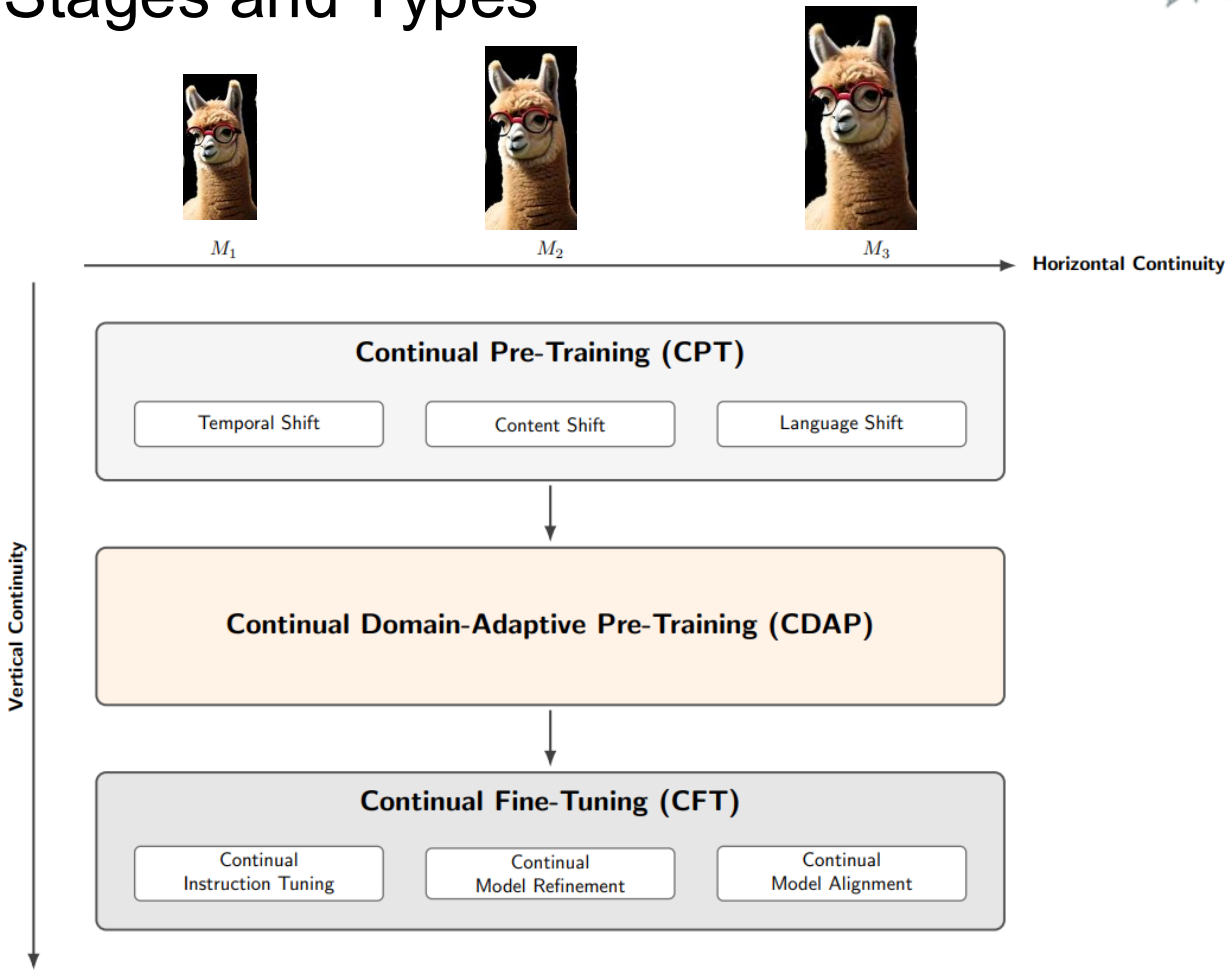


(a) Vertical Continual Learning of LLMs

Task Heterogeneity. Stemming from the inherent disparity between the formulation of upstream tasks and downstream task

Inaccessible UPstream Data : Data collected and curated under diferent protocols may not be accessible to some downstream entities.

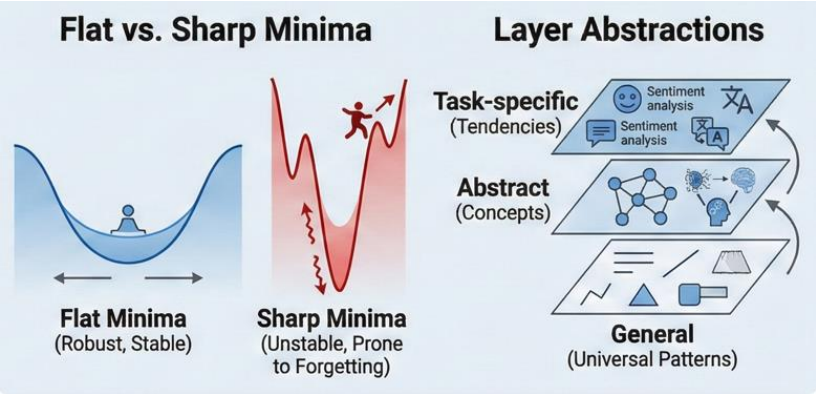
CL in LLMs : Stages and Types



Continual Learning in LLMs

Pre-training places LLMs in a broad basin of parameter space

- Trained on huge diversity: many domains styles, modalities, tasks
- Leads to general reusable features
- Training diversity forces optimization into stable regions not task-specific minimas
- Flat minima is robust to small parameter perturbations; less catastrophic forgetting
- Small updates during CL stays inside the same basin; easier knowledge retention



Why LLM CL Works Better

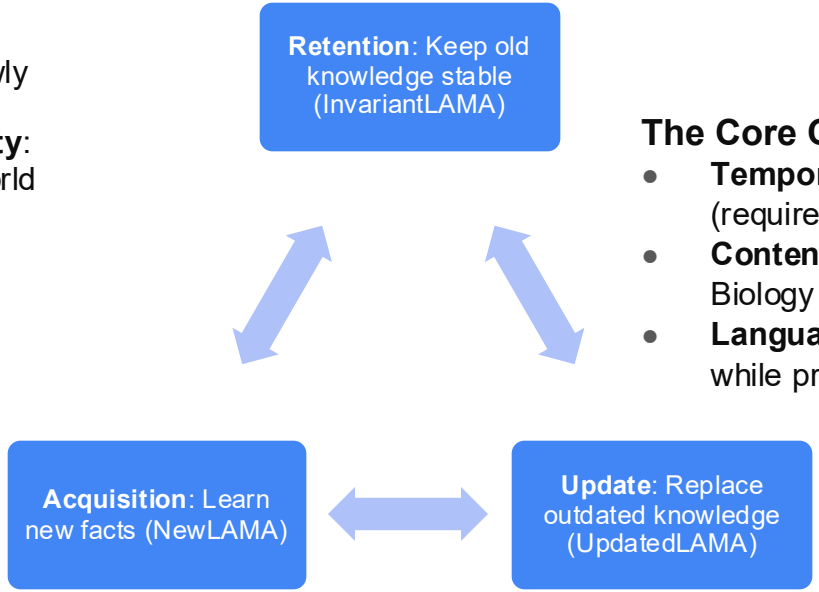
Pre-training diversity (Many domains, tasks) → **Broad Basin** (Stable Parameter Space) → **Robustness** (Less Catastrophic forgetting) → **Easier Continual Learning** (Fine-tuning within stable Foundation)

Result: LLMs start in a robust region of parameter space, so CL mostly fine-tunes within a stable foundation rather than learning from scratch

Continual Pre-Training: Adapting LLMs dynamically to new data

What CPT Is

- Continuous pre-training on newly collected data (supplier side)
- Top layer of **Vertical Continuity**:
- Maintains alignment as real-world distributions evolve



The Core Challenge: Distribution Shifts

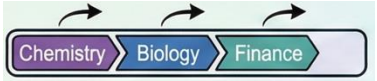
- **Temporal Shift**: New facts replace old ones (requires update + retention)
- **Content Shift**: New Domains (Chemistry – Biology – Finance)
- **Language Shift**: Acquiring new languages while preserving old ones.

Why CPT Is Needed

- Static LLMs become outdated: Temporal Misalignment
- Avoids costly “retrain-from-scratch”
- CPT updates timestamp-sensitive, domain-shifting, and multilingual knowledge.

Distributional Shifts in CPT: Content Shift and Temporal Shift

Content Shift is Common and Beneficial but Challenging



Sequentially Learning new fields

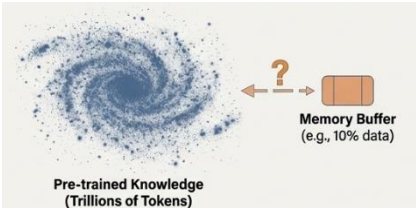
- Sequentially learning new fields (e.g., chemistry → biology → finance) requires **domain-specialization without forgetting** earlier content.
- Content shifts require balancing rapid domain adaptation with minimal interference to prior knowledge.

Why Temporal Shift Is Uniquely Challenging

- Temporal shifts introduce **conflicting facts** over time.

- Unlike content or language shift, we **don't want to retain** outdated facts, we need to update them.
- struggles when tasks contain *mutually inconsistent knowledge*.

Why Replay fails In CPT



- The sheer volume of pre-training corpora makes replay buffers **computationally unrealistic**.
- Replay-based methods struggle due to **conflicting facts** from different time stamps
- At LLM scale, replay becomes **infeasible** and even **slows adaptation** to new domains.

Qin, Yujia, et al. "Recyclable tuning for continual pre-training." arXiv preprint arXiv:2305.08702 (2023).
Gururangan, Suchin, et al. "Demix layers: Disentangling domains for modular language modeling."
Chen, Wuyang, et al. "Lifelong language pretraining with distribution-specialized experts." International Conference on Machine Learning. PMLR, 2023.
J. Jang, S. Ye, S. Yang, J. Shin, J. Han, G. Kim, S. J. Choi, and M. Seo. Towards continual knowledge learning of language models. In ICLR
J. Jang, S. Ye, C. Lee, S. Yang, J. Shin, J. Han, G. Kim, and M. Seo. Temporalwiki: A lifelong benchmark for training and evaluating ever-evolvinglanguage models. 2022.

Distributional Shifts in CPT: Language Shift

Forgetting Patterns Vary across Languages

For a LLM initially trained on English corpora:


Danish

Almost No Forgetting; Sometimes positive backward transfer


Norwegian

Mostly mild, sometimes positive, sometimes slight forgetting.


Icelandic

Consistent catastrophic forgetting of previously learned languages

Larger LLMs lower overall loss however the trend remains the same

Why Language Shift is Hard to Solve

Forgetting is not trivial; Simple freezing, partial freezing or naïve mixing strategies are not sufficient.

[63] E. Gogoulou, T. Lesort, M. Boman, and J. Nivre. Continual learning under language shift, 2024.
[118] C.-A. Li and H.-Y. Lee. Examining forgetting in continual pre-training of aligned large language models, 2024

CPT : DEMIX Layers for Disentangling Domains for Modular Language Modeling

Motivation:
Monolithic LLMs Struggle with adaptation and control

Standard “dense training” updates all parameters on all data, leading to significant catastrophic forgetting

DEMIX Layers : Idea

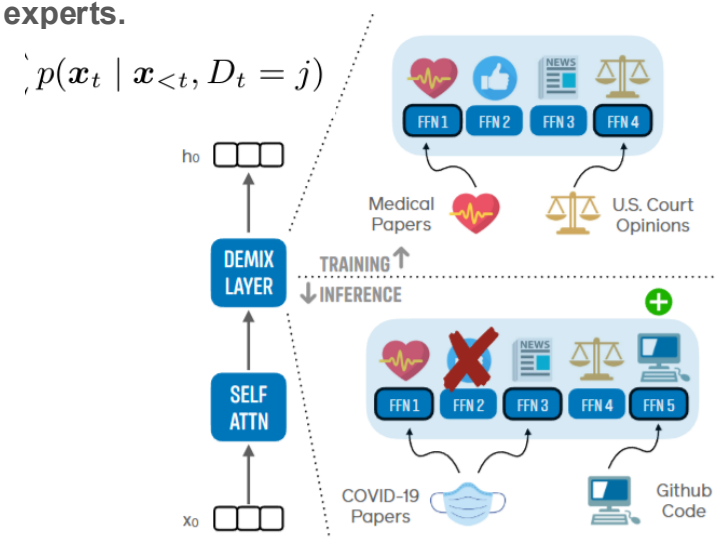
- Most domain-specific and factual knowledge in Transformers is stored in FFN layers, not attention layers.
- DEMix layers implement domain-conditioned FFN experts, trained incrementally and frozen across CPT steps, yielding architectural guarantees against catastrophic forgetting

$$\mathrm{FFN}_{\mathrm{DEMix}}(h) = \sum_{k}^{K} g_k(d) \cdot \mathrm{FFN}^{(k)}(h)$$

where:

- k indexes **domains** (news, biomedical, code, etc.)
- $\mathrm{FFN}^{(k)}$ is the FFN expert for domain k
- $g_k(d)$ is a **domain-conditioned gate**

The Methodology:
Replace monolithic FFNs with modular, domain-specific experts.



- **Key Feature:** This creates a modular architecture where domain-specific knowledge is disentangled into separated components, while shared parameters (like attention layers) are still learned across all data

Gururangan, Suchin, et al. "Demix layers: Disentangling domains for modular language modeling." Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2022.

CPT : DEMIX Layers for Disentangling Domains for Modular Language Modeling

Results: Dynamic expert mixing leads to better generalization

- Uses GPT3 model and perplexity metric

	Domain	Corpus
TRAINING	1B	30M NewsWire sentences (Chelba et al., 2014)
	CS	1.89M full-text CS papers from S2ORC (Lo et al., 2020)
	LEGAL	2.22M U.S. court opinions, 1658 to 2018 (Caselaw Access Project, 2018)
	MED	3.2M full-text medical papers from S2ORC (Lo et al., 2020)
	WEBTEXT†	8M Web documents (Gokaslan and Cohen, 2019)
	REALNEWS†	35M articles from REALNEWS (Zellers et al., 2019)
	REDDIT	Reddit comments from pushshift.io (Baumgartner et al., 2020)
	REVIEWS†	30M Amazon product reviews (Ni et al., 2019)

Domain	1.3B parameters per GPU		
	DENSE	DEMIX (naive)	DEMIX (cached prior; §5.4)
1B	11.8	11.5	**11.3**
CS	13.5	12.2	**12.1**
LEGAL	6.8	**6.7**	**6.7**
MED	9.5	9.2	**9.1**
WEBTEXT	**13.8**	14.6	14.3
REALNEWS	**12.5**	13.3	13.1
REDDIT	28.4	30.6	**28.1**
REVIEWS	14.0	12.6	**12.5**
Average	13.8	13.8	**13.4**

Add

Adapt to new domains without catastrophic forgetting by adding a new expert and training only it's parameters and keeping the rest of the model frozen.

Remove

Control model behaviour by disabling domains. Simply deactivate a expert at inference time to simulate a model never trained on that data. Offering a lightweight mechanism to restrict access to unwanted data.

Novel/Unseen Domains ?

Gururangan, Suchin, et al. "Demix layers: Disentangling domains for modular language modeling." Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2022.

CPT : DEMIX Layers for Disentangling Domains for Modular Language Modeling

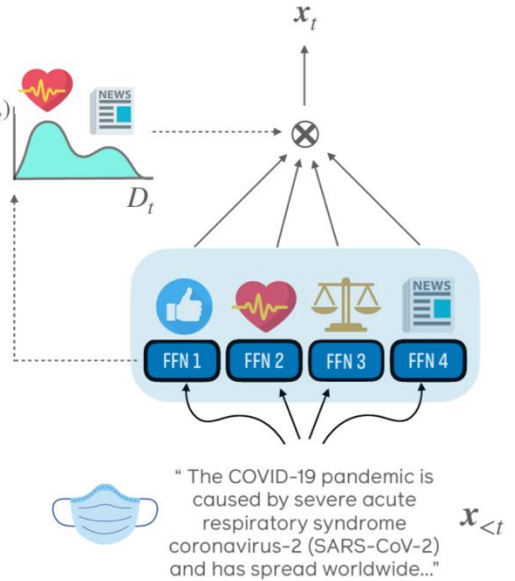
Results: Dynamic expert mixing leads to better generalization

- Uses GPT3 model and perplexity metric
- At inference time, we can form a parameter-free, weighted ensemble of experts to handle data from unseen or heterogeneous domains. This is done by estimating a posterior probability over domains for the input tex‘

$$p(D_t = j \mid \boldsymbol{x}_{<t}) = \frac{p(\boldsymbol{x}_{<t} \mid D_t = j) \cdot p(D_t = j)}{p(\boldsymbol{x}_{<t})} \quad (5$$

$$= \frac{p(\boldsymbol{x}_{<t} \mid D_t = j) \cdot p(D_t = j)}{\sum_{j'=1}^{n} p(\boldsymbol{x}_{<t} \mid D_t = j') \cdot p(D_t = j')}$$

	Parameters per GPU			
	125M	350M	760M	1.3B
DENSE	25.9	21.4	18.4	17.8
DENSE (balanced)	25.3	19.6	18.3	17.1
+DOMAIN-TOKEN	24.8	20.4	18.4	18.0
DEMIX (naive)	28.8	23.8	21.8	21.1
DEMIX (average)	27.2	22.4	21.5	20.1
DEMIX (uniform)	24.5	20.5	19.6	18.7
DEMIX (updating)	21.9	18.7	17.6	17.1
DEMIX (cached)	**21.4**	**18.3**	**17.4**	**17.0**



Mix

Handle Heterogenous and unseen domains by dynamically combining experts using a Bayesian approach at test time

Drawbacks ?

Gururangan, Suchin, et al. "Demix layers: Disentangling domains for modular language modeling." Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2022.

CPT : DEMIX Layers for Disentangling Domains for Modular Language Modeling

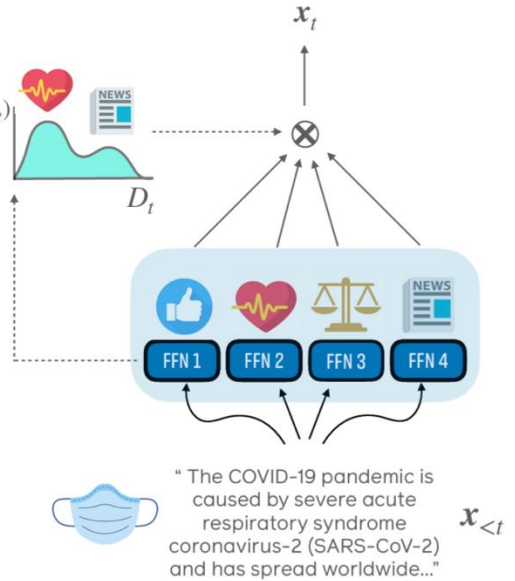
Results: Dynamic expert mixing leads to better generalization

- Uses GPT3 model and perplexity metric
- At inference time, we can form a parameter-free, weighted ensemble of experts to handle data from unseen or heterogeneous domains. This is done by estimating a posterior probability over domains for the input tex‘

$$p(D_t = j \mid \boldsymbol{x}_{<t}) = \frac{p(\boldsymbol{x}_{<t} \mid D_t = j) \cdot p(D_t = j)}{p(\boldsymbol{x}_{<t})} \quad (5$$

$$= \frac{p(\boldsymbol{x}_{<t} \mid D_t = j) \cdot p(D_t = j)}{\sum_{j'=1}^{n} p(\boldsymbol{x}_{<t} \mid D_t = j') \cdot p(D_t = j')}$$

	Parameters per GPU			
	125M	350M	760M	1.3B
DENSE	25.9	21.4	18.4	17.8
DENSE (balanced)	25.3	19.6	18.3	17.1
+DOMAIN-TOKEN	24.8	20.4	18.4	18.0
DEMIX (naive)	28.8	23.8	21.8	21.1
DEMIX (average)	27.2	22.4	21.5	20.1
DEMIX (uniform)	24.5	20.5	19.6	18.7
DEMIX (updating)	21.9	18.7	17.6	17.1
DEMIX (cached)	**21.4**	**18.3**	**17.4**	**17.0**



Mix

Handle Heterogenous and unseen domains by dynamically combining experts using a Bayesian approach at test time

Drawbacks

- No knowledge transfer across domains
- Shared components drift towards later domain

Gururangan, Suchin, et al. "Demix layers: Disentangling domains for modular language modeling." Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2022.

CPT : Lifelong Language Pretraining with Distribution-Specialized Experts

Methodology: Addresses drawbacks of DEMIX - no explicit domain labels, regularization over shared parameters

The methodology is balanced on three pillars.

- **Progressive Expert Expansion**: The MoE architecture's capacity in dynamically increased by adding new, specialized experts for each new data distribution. This avoids overwriting existing parameters.

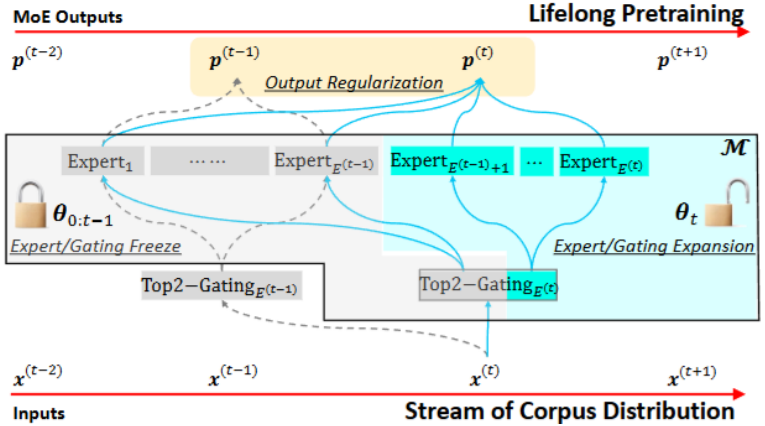
$$\mathrm{FFN}_{\mathrm{MoE}}(h) = \sum_{k=1}^{K} g_k(h)\,\mathrm{FFN}^{(k)}(h)$$

where:

- $\mathrm{FFN}^{(k)}$: expert k
- $g_k(h)$: **learned gating function**
- Routing is **token-level**, not domain-level

Routing depends on **hidden representation**

$$g(h) = \mathrm{Softmax}(W_g h)$$



- **Knowledge Preservation via Freezing**: Previously trained experts and their corresponding gating layers are frozen. This explicitly protects learned knowledge from being erased during training on new data.
- **Implicit Regularization via Distillation**: Shared components of the network (e.g., attention layers) are guided using knowledge distillation from the model's previous state. This prevents them from drifting too far while still allowing adaptation to new data.

Chen, Wuyang, et al. "Lifelong language pretraining with distribution-specialized experts." International Conference on Machine Learning. PMLR, 2023.

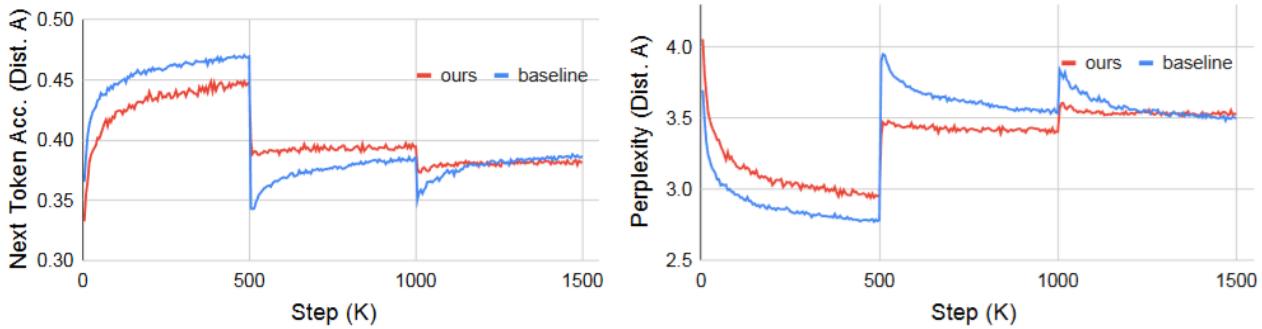
CPT : Lifelong-MoE Results

LLM and MoE based on GLAM. Experiments conducted on 3 different domains

Distribution	Corpus	Tokens (B)
$\mathcal{A}$	Wikipedia (19%) Filtered Webpages (81%)	3 143
$\mathcal{B}$	i18n	366
$\mathcal{C}$	Conversations	174

Lifelong-MoE maintains performance on previously learned distributions, directly preventing the catastrophic forgetting observed in standard sequential pretraining

Next-Token Accuracy and Perplexity on Distribution A (Each distribution is trained for 500K



Lifelong-MoE not only dramatically reduces forgetting buy also achieves competitive or state of the art performance outperforming strong baselines like Memory Replay and even a Dense Oracle model trained jointly on all data forspecifc tasks.

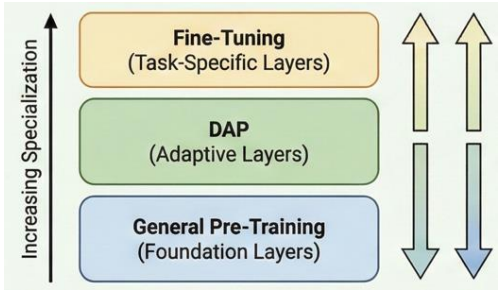
Domain Adaptive Pre-Training (DAPT)

What DAP is & Why it matters

Improves performance on specialized
(e.g., Legal, Medical Financial)

How DAP fits into Vertical Continuity

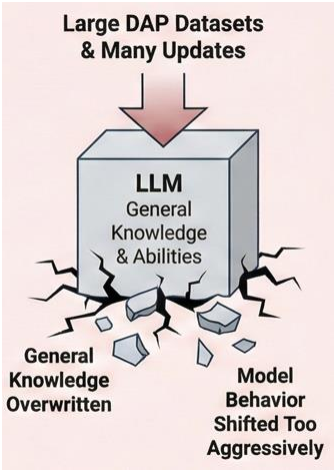
General pretraining **underrepresents** specialized domains. DAPT corrects this distributional mismatch.



Updating with New Domain Knowledge

Preserving General-Purpose Abilities (Reasoning, Language, Commonsense)

The Core Challenge: Vertical Forgetting



- DAP usually damages performance on general tasks (safety, reasoning & Instruction-Following.

Key Observations From the Literature

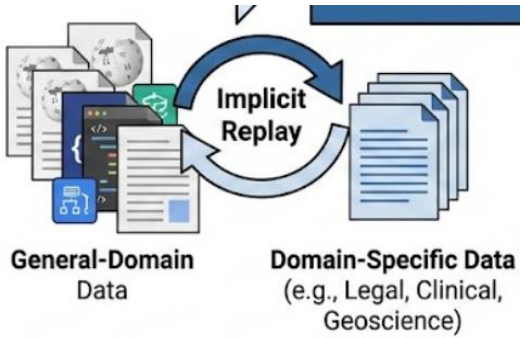
- **DAP is usually a single-stage process**
  Multi-stage or Continual DAP is rare.
- **DAP is already being treated as CL**
  CL concepts are used without naming them: replay = “data mixing”, adapters = “domain adapters”
- **Diversity of CL techniques used in DAP is limited**

Domain Adaptive Pre-Training: CL Techniques

Replay appears Implicitly in DAP Pipelines

Key Observations:
This implicitly functions as **experience replay**
Goal: **reduce vertical forgetting** of general capabilities

Example	Replay mechanism	Mixing amount / ratio
SaulLM	Mixes a small amount of general data into legal-domain DAP	2% general-domain
Me-LLaMA	Mixes general data when adapting to clinical/biomedical domains	~25% general-domain
HuatuoGPT-II	Uses priority sampling to avoid issues from fixed-rate mixing	Priority sampling (no fixed %)
GeoGalactica	Mixes geoscience (52B tokens), arXiv papers, and code data	8:1:1 (Geoscience:arXiv:Code Data)



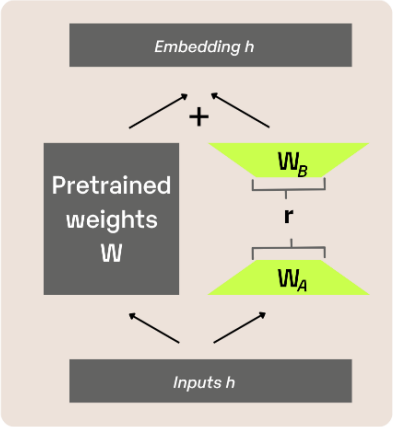
P. Colombo, T. P. Pires, M. Boudiaf, D. Culver, R. Melo, C. Corro, A. F. T. Martins, F. Esposito, V. L. Raposo, S. Morgado, and M. Desa. Saullm-7b: A pioneering large language model for law, 2024.
J. Chen, X. Wang, A. Gao, F. Jiang, S. Chen, H. Zhang, D. Song, W. Xie, C. Kong, J. Li, X. Wan, H. Li, and B. Wang. Huatuogpt-ii, one-stage training for medical adaption of llms. CoRR, abs/2311.09774, 2023
Z. Lin, C. Deng, L. Zhou, T. Zhang, Y. Xu, Y. Xu, Z. He, Y. Shi, B. Dai, Y. Song, B. Zeng, Q. Chen, T. Shi, T. Huang, Y. Xu, S. Wang, L. Fu, W. Zhang, J. He, C. Ma, Y. Zhu, X. Wang, and C. Zhou. Geogalactica: A scientific large language model in geoscience, 2023
Q. Xie, Q. Chen, A. Chen, C. Peng, Y. Hu, F. Lin, X. Peng, J. Huang, J. Zhang, V. Keloth, et al. Me llama: Foundation large language models for medical applications. arXiv preprint arXiv:2402.12749, 2024.

Continual Domain Adaptive Pre-Training:

Adapter-Based & LoRA-Based DAP (Architecture Expansion)

- Architecture expansion adds new task capacity, preserves general-domain representations, and limits interference, reducing catastrophic forgetting.
- **Adapter-based DAP (e.g., AF Adapter)** expands attention & FFN widths, with only adapters trained for stronger vertical forgetting resistance.
- **LoRA-based DAP (e.g., Hippocrates)** injects medical knowledge via LoRA, preserving general reasoning abilities.
- **IRCoder** applies LoRA on compiler IR representations, improving multilingual code instruction following.
- **LLaMA Pro** adds multiple identity copies of transformer blocks for stronger forgetting resistance through later tuning compared to vanilla LoRA.

Low-Rank Adaptation(LoRA)



Key Takeaway:

- Replay, Adapters/LoRA, and Regularization all act as **continual learning mechanisms**—often unintentionally—to prevent vertical forgetting during DAP.
- This shows that **DAP is already a CL process**, requiring deliberate strategies to preserve general-purpose LLM capabilities.

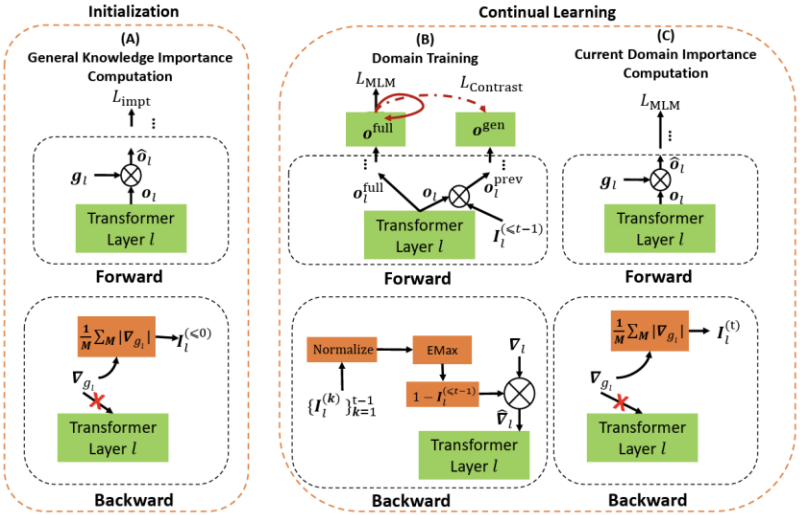
Y. Yan, K. Xue, X. Shi, Q. Ye, J. Liu, and T. Ruan. Af adapter: Continual pretraining for building chinese biomedical language model. In 2023 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), pages 953–957, Los Alamitos, CA, USA, dec 2023. IEEE Computer Society.

E. C. Acikgoz, O. B. İnce, R. Bench, A. A. Boz, İ. Kesen, A. Erdem, and E. Erdem. Hippocrates: An open-source framework for advancing large language models in healthcare. arXiv preprint arXiv:2404.16621, 2024.

I. Paul, J. Luo, G. Glavaš, and I. Gurevych. Ircoder: Intermediate representations make language models robust multilingual code generators, 2024. [178] A. Pentina. Theoretical foundations of multi-task lifelong learning. PhD thesis, 2016.

DAPT : Continual DA-pre-training of LMs with Soft-masking

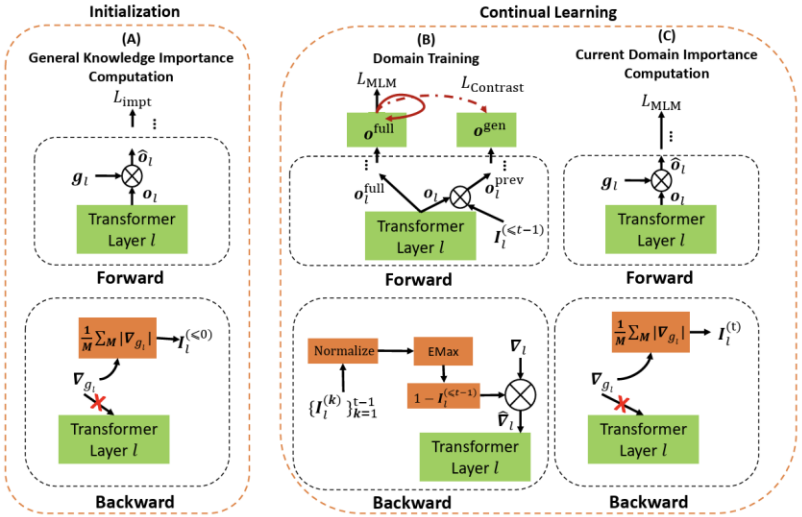
- LLM pre-trained on D0, we incrementally DAP-train a sequence of domain corpora D1,D2 .
  - overcome CF for each new domain and general language knowledge in the LM.
  - encourage knowledge transfer (KT) across domains
- DAS (Continual DA-pre-training of LMs with Soft-masking).
  - novel **soft-masking mechanism** that computes the importance of units for general or domain knowledge
  - Continual learning uses soft-masking to prevent CF on the general and domain knowledge
  - Initialization computes the importance of units to the general knowledge without pre-training data (D0).



Ke, Shao, Lin, Konishi, Kim, & Liu (ICLR 2023): “Continual Pre-training of Language Models.

DAPT : Continual DA-pre-training of LMs with Soft-masking

- LLM pre-trained on D0, we incrementally DAP-train a sequence of domain corpora D1,D2 .
  - overcome CF for each new domain and general language knowledge in the LM.
  - encourage knowledge transfer (KT) across domains
- DAS (Continual DA-pre-training of LMs with Soft-masking).
  - novel **soft-masking mechanism** that computes the importance of units for general or domain knowledge
  - Continual learning uses soft-masking to prevent CF on the general and domain knowledge
  - **Initialization** computes the importance of units to the general knowledge without pre-training data (D0).
  - **domain training** takes the importance scores accumulated so far and learn from the input data of the current domain
  - **importance computation** computes the importance scores for the current domain
  - **Proxy KL-divergence loss**: Detect units that are important for LM's robustness based on 2 input passes
  - **Accumulating importance.**



$$\boldsymbol{I}_l = \frac{1}{N}\sum_{n=1}^{N}\left|\frac{\partial \mathcal{L}_{\text{impt}}(\boldsymbol{x}_n, y_n))}{\partial \boldsymbol{g}_l}\right|,$$

$$\mathcal{L}_{\text{impt}} = \text{KL}(f^1_{\text{LM}}(\boldsymbol{x}^{\text{sub}}_n), f^2_{\text{LM}}(\boldsymbol{x}^{\text{sub}}_n)),$$

$$\boldsymbol{I}^{(\leq t-1)}_l = \text{EMax}(\{\boldsymbol{I}^{(t-1)}_l, \boldsymbol{I}^{(\leq t-2)}_l\})|,$$

Ke, Shao, Lin, Konishi, Kim, & Liu (ICLR 2023): "Continual Pre-training of Language Models.

DAPT : Continual DA-pre-training of LMs with Soft-masking

- 6 unlabeled domain corpora for DAP-training and Roberta as LM

Unlabelde Domain Datasets			End-Task Classification Datasets				
Source	Dataset	Size	Dataset	Task	#Training	#Testing	#Classes
Reviews	Yelp Restaurant	758MB	Restaurant	Aspect Sentiment Classification (ASC)	3,452	1,120	3
	Amazon Phone	724MB	Phone	Aspect Sentiment Classification (ASC)	239	553	2
	Amazon Camera	319MB	Camera	Aspect Sentiment Classification (ASC)	230	626	2
Academic Papers	ACL Papers	867MB	ACL	Citation Intent Classification	1,520	421	6
	AI Papers	507MB	AI	Relation Classification	2,260	2,388	7
	PubMed Papers	989MB	PubMed	Chemical-protein Interaction Prediction	2,667	7,398	13

Category	Domain	Restaurant		ACL		AI		Phone		PubMed	Camera		Average	
	Model	MF1	Acc	MF1	Acc	MF1	Acc	MF1	Acc	MF1	MF1	Acc	MF1	Acc
Non-CL	Pool	80.96	87.80	69.69	74.11	68.55	75.97	84.96	86.95	73.34	86.03	90.83	77.25	81.50
	RoBERTa	79.81	87.00	66.11	71.26	60.98	71.85	83.75	86.08	72.38	78.82	87.03	73.64	79.27
	DAP-RoBERTa	80.84	**87.68**	68.75	73.44	68.97	75.95	82.59	85.50	72.84	84.39	89.90	76.40	80.89
	DAP-Adapter	80.19	87.14	68.87	72.92	60.55	71.38	82.71	85.35	71.68	83.62	89.23	74.60	79.62
	DAP-Prompt	79.00	86.45	66.66	71.35	61.47	72.36	84.17	86.53	**73.09**	85.52	90.38	74.98	80.03
CL DAP-train	NCL	79.52	86.54	68.39	72.87	67.94	75.71	84.10	86.33	72.49	85.71	90.70	76.36	80.77
	NCL-Adapter	80.13	87.05	67.39	72.30	57.71	69.87	83.32	85.86	72.07	83.70	89.71	74.05	79.48
	DEMIX	79.99	87.12	68.46	72.73	63.35	72.86	78.07	82.42	71.73	86.59	91.12	74.70	79.66
	BCL	78.97	86.52	**70.71**	**74.58**	66.26	74.55	81.70	84.63	71.99	85.06	90.51	75.78	80.46
	CLASSIC	79.89	87.05	67.30	72.11	59.84	71.08	84.02	86.22	69.83	86.93	91.25	74.63	79.59
	KD	78.05	85.59	69.17	73.73	67.49	75.09	82.12	84.99	72.28	81.91	88.69	75.17	80.06
	EWC	**80.98**	87.64	65.94	71.17	65.04	73.58	82.32	85.13	71.43	83.35	89.14	74.84	79.68
	DER++	79.00	86.46	67.20	72.16	63.96	73.54	83.22	85.61	72.58	87.10	91.47	75.51	80.30
	HAT	76.42	85.16	60.70	68.79	47.37	65.69	72.33	79.13	69.97	74.04	85.14	66.80	75.65
	HAT-All	74.94	83.93	52.08	63.94	34.16	56.07	64.71	74.43	68.14	65.54	81.44	59.93	71.33
	HAT-Adapter	79.29	86.70	68.25	72.87	64.84	73.67	81.44	84.56	71.61	82.37	89.27	74.63	79.78
	DAS	80.34	87.16	69.36	74.01	**70.93**	**77.46**	**85.99**	**87.70**	72.80	**88.16**	**92.30**	**77.93**	**81.91**

Continual Fine-Tuning (CFT): Bottom Layer of Vertical Continuity

What CFT Is

- CFT lies at the **bottom layer** of the vertical continuity hierarchy.
- The model is trained on **successive homogeneous tasks** drawn from an evolving (but relatively stable) data distribution.

Why CFT is easier than CPT/DAP

CPT/ DAP (Global Continual Alignment)

Balance General Knowledge Retention vs. Domain Injection

CFT (Local Improvement)

Focus on Direct Task Adaptation

- Lower Catastrophic forgetting due to more homogeneous tasks
- Simpler optimization and CL constraints

Key Sub Areas of CFT

Continual Instruction Tuning (CIT)

Sequential instruction datasets, improving reasoning & alignment

Continual Model Refinement (CMR)

Ongoing Improvement of base models with updated data

Continual Model Alignment (CMA)

RLHF/DPO style updates done sequentially for safety and trustworthiness.

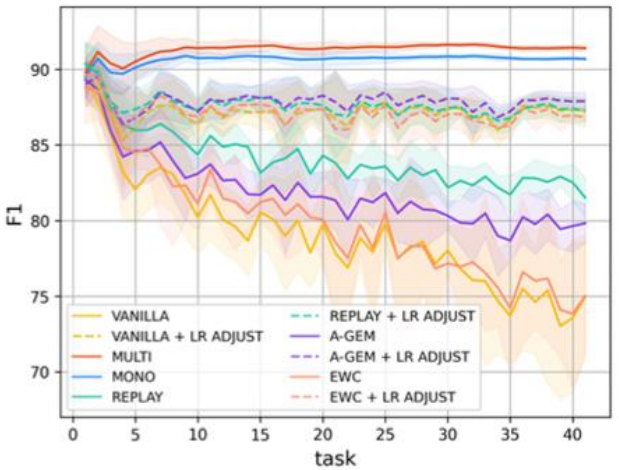
- CL techniques in LLM are most actively explored in CFT.
- CL techniques such as Regularization, parameter Isolation and replay style sampling are widely used and effective.

CFT : LR ADJUST

- Approaches leverage the inherent anti-forgetting nature of LLMs while avoiding the adoption of overly complex CL techniques
- LR ADJUST [255] proposes dynamically adjusting the learning rate to mitigate the overwriting of knowledge from new languages onto old ones.
- Lower base LR for later tasks, Task-dependent LR scaling, Layer-wise LR decay, where Lower (foundational) layers receive smaller updates and Higher (task-specific) layers adapt more aggressively
- Limitations : Cannot fully prevent forgetting under strong domain/task shifts, Sensitive to LR scheduling choices

1: Randomly initialize the classifier on θ
2: **for all** $D_t \in \mathcal{D}$ **do**
3: Adjust learning rate to lr_t
$lr_t = \max(lr_{\min}, lr_{t-1} \cdot \gamma)$
4: Compute $\nabla_\theta \mathcal{L}_{D_t}(f_\theta)$ using D_t
5: $\theta_{t+1} \leftarrow \theta_t - lr_t \nabla_\theta \mathcal{L}_{D_t}(f_\theta)$
6: **end for**

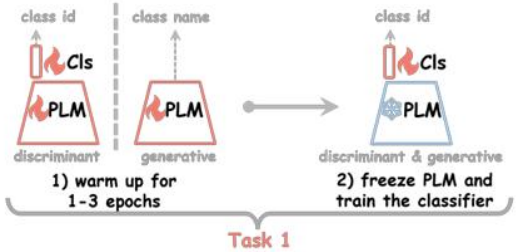
multilingual natural language understanding data set, MASSIVE (FitzGerald et al., 2022)



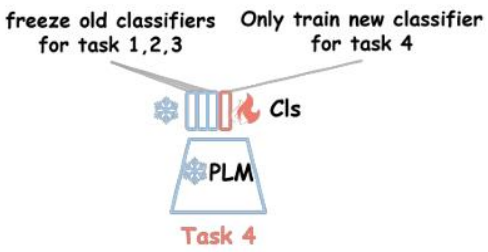
Winata, G. I., Xie, L., Radhakrishnan, K., Wu, S., Jin, X., Cheng, P., Kulkarni, M., & Preotiuc-Pietro, D. (2023). Overcoming catastrophic forgetting in massively multilingual continual learning. *Findings of the Association for Computational Linguistics: ACL 2023*,

CFT : SEQ/SEQ*

Some works[1,2] introduce strategies for fine-tuning LLMs on a sequence of **downstream classification tasks**, such as freezing the LLM and old classifier's parameters after warm-up (LLM and classifier are trained jointly on the *first* task for a short period), and allocating new classifier head for future tasks.

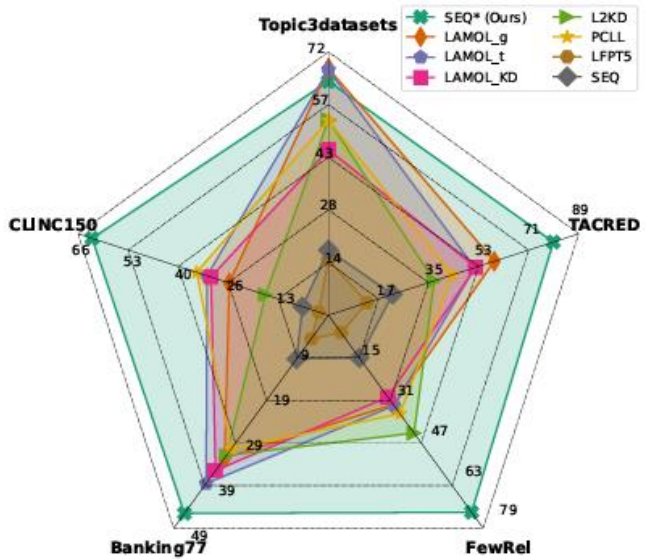


(a) S1: warm-up and freeze PLM



(b) S2: freeze old classifiers

Granularity	Task	Dataset	# Classes	# Tasks
Sentence Level	Text Classification	Topic3Datasets	25	5
	Intent Classification	CLINC150	150	15
		Banking77	77	7
	Relation Extraction	FewRel	80	8
		TACRED	40	8
Word Level	Named Entity Recognition	Few-NERD	66	11
		Ontonotes5	18	6
		I2B2	16	5



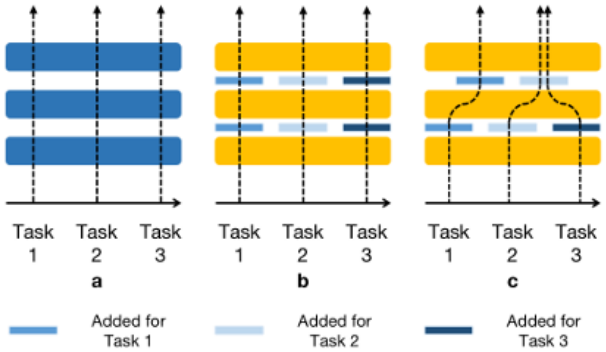
J. Zheng, S. Qiu, and Q. Ma. Learn or recall? revisiting incremental learning with pre-trained language models, 2023.
Zhang et. Al. Pythia: A suite for analyzing large language models across training and scaling, ICML 2023

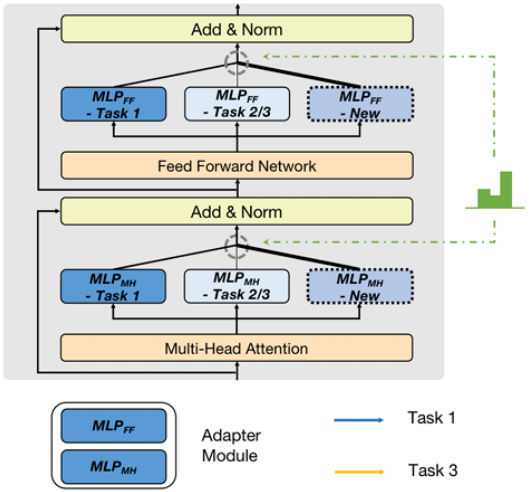
CFT: Compositional Adapters

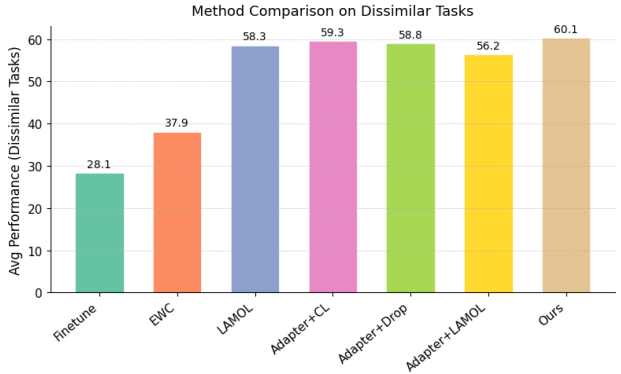
A method for adaptively adding **compositional modules (adapters)** during continual sequence generation tasks. Before training on new domains, a decision stage determines which trained module can be reused.

Hidden state mixing: For each transformer layer, the model evaluates all existing modules (from previous tasks) and a newly added module.

- Combines output of each module using a **weighted average** of their outputs
- Model learns weight coefficients and newly added module
- The module with the **highest learned weight** is considered the most useful and is selected for reuse.







GPT-2 model on data sets WikiSQL (SQL query generation), CNN/DailyMail (news article summarization) MultiWOZ (semantic state sequence generation

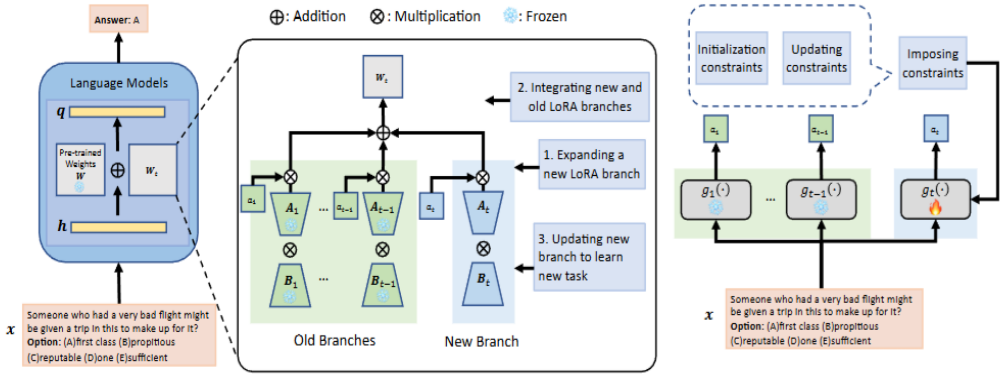
Y.Zhang,X.Wang,andD.Yang. Continual sequence generation with adaptive compositional modules. In S. Muresan, P. Nakov, and A. Villavicencio,, ACL 2022

CFT : Gated Integration of Low-Rank Adaptation for Continual Learning(GainLoRA)

- In several practical situations, actual task ID is not known.
- Existing solutions simply add all the separate LoRA branches together to form the final output (Model Merging), causing uncontrollable inference.
- GainLoRA - introduces gating modules to integrate the new and old LoRA branches.

Core Innovation: Task-wise Gating Introduces independent gating modules for each task. Each gate outputs a learned integration coefficient that determines the influence of the i-th LoRA branch

$$\boldsymbol{W}_t = \boldsymbol{W}_{t-1} + a_t \boldsymbol{A}_t \boldsymbol{B}_t = \sum_{i=1}^{t} a_i \boldsymbol{A}_i \boldsymbol{B}_i, \quad a_i \in [0,1], \quad a_i \sim 0;\ i \in [0, t-1], \quad a_i \sim 1;\ i = T \qquad a_i = g_i(\boldsymbol{x})$$



(a) Expandable LoRA Architecture in GainLoRA

(b) Gating Modules in GainLoRA

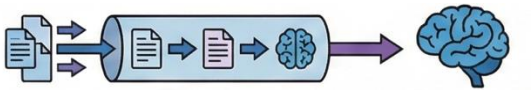
Performance evaluation of SuperNI benchmark - dialogue generation, information extraction, question answering, summarization, sentiment analys

Models	Methods	Order 1 AP↑	Order 1 FT↓
Llama-2-7B	O-LoRA [61]	39.37	15.84
	GainLoRA (O-LoRA)	51.10	4.96
	InfLoRA [32]	42.93	11.23
	GainLoRA (InfLoRA)	**51.27**	**2.84**
Llama-2-13B	O-LoRA [61]	43.92	14.15
	GainLoRA (O-LoRA)	52.47	4.78
	InfLoRA [32]	43.64	14.85
	GainLoRA (InfLoRA)	**53.64**	**2.87**

Liang, Yan-Shuo, and Wu-Jun Li. *Gated Integration of Low-Rank Adaptation for Continual Learning of Language Models (GainLoRA) NeurIPS 2025*

Continual Instruction Tuning (CIT): Learning from Evolving Instruction Streams

What is CIT?



When Instruction datasets arrive as a stream. Requires learning new instructions without forgetting the previous ones.

Why CIT is Unique?

- Instruction data is *sequential and heterogeneous. (*new tasks, formats, and reasoning patterns.)
- Forgetting manifests as reduced instruction-following ability ("half-listening" effect).
- rich natural language instructions enable knowledge transfer and reduce forgetting,

Each instance d_τ in the task τ forms a triple(i,c,y)

Key Techniques Used in CIT

Replay Based Methods

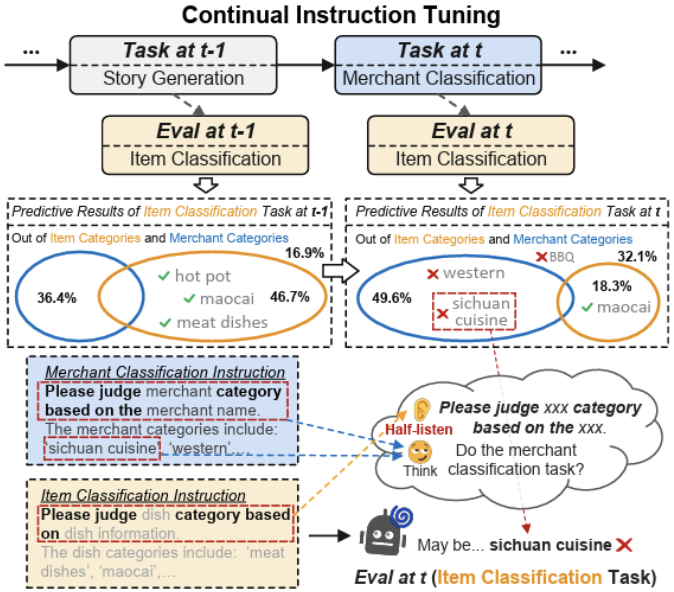
- **KPIG Replay:** Uses Key Part Information Gain for dynamic replay selection, reducing “half listening”.
- **SSR**: Uses LLM to generate synthetic replay data, lowering compute cost while preserving skills.

PEFT-Based Approaches

- **O-LoRA** Learns new instructions in orthogonal subspace, preserving older LoRA weights; minimizes interference.
- **SAPT**: Shared attentive learning + selection module to manage knowledge transfer and forgetting jointly.

CIT : **Key-Part Information Gain (KPIG)**

- **Replay also introduces semantic interference, and the model begins to shortcut to the lowest-effort interpretation , following only primary verb and ignoring modifiers.**
- **Half-Listening Problem** — where the model partially follows instructions because prior replay or fine-tuning emphasized only fragments of meaning.
- **Key parts are consecutive spans in the instruction which provide task-aware guidance on the content, length, and format to generate desired responses..**
- **Improve replay efficiency by computing Key-Part Information Gain (KPIG) on masked parts to dynamically select replay data, addressing the "half-listening" issue in instruction following.**



Y. He, X. Huang, M. Tang, L. Meng, X. Li, W. Lin, W. Zhang, and Y. Gao. Don't half-listen: Capturing key-part information in continual instruction tuning, ACL 2024.

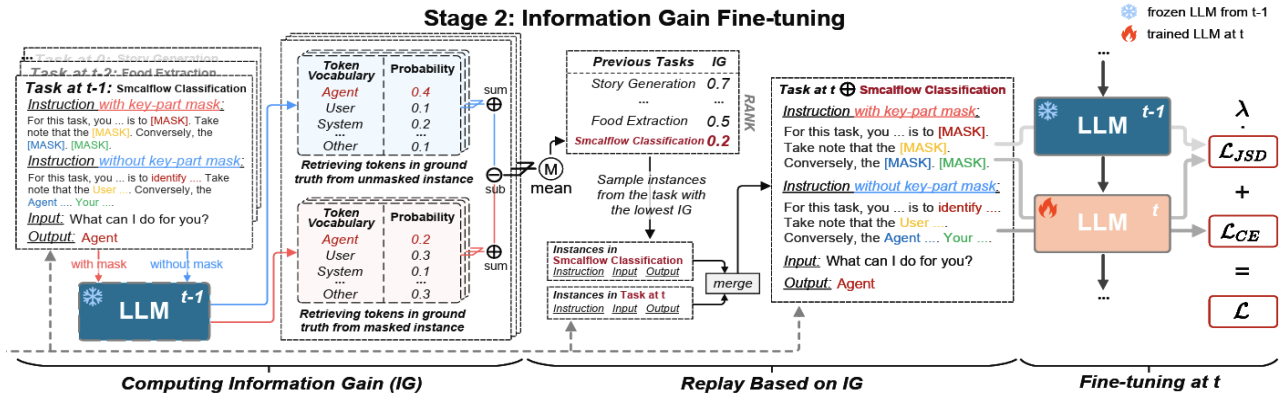
CIT : **Key-Part Information Gain (KPIG)**

- **KPIG (Key-Part Information Gain)**This is used to rank training or replay examples based on how much **critical semantic information** they carry

- Masking *important phrase spans* in the instruction (e.g., goals, constraints, modifiers).Measuring how much the model's confidence drops when these masked parts are removed.

$$\mathcal{G}(d_\tau, d_\tau^m) = \mathrm{Info}(y|x) - \mathrm{Info}\big(y|\mathrm{mask}(x)\big)$$

- Select M seen tasks with the lowest mean IG as replay tasks and choose MN samples for reply : Low information gain indicates that the model is not effectively capturing the task-specific constraints in the instruction.



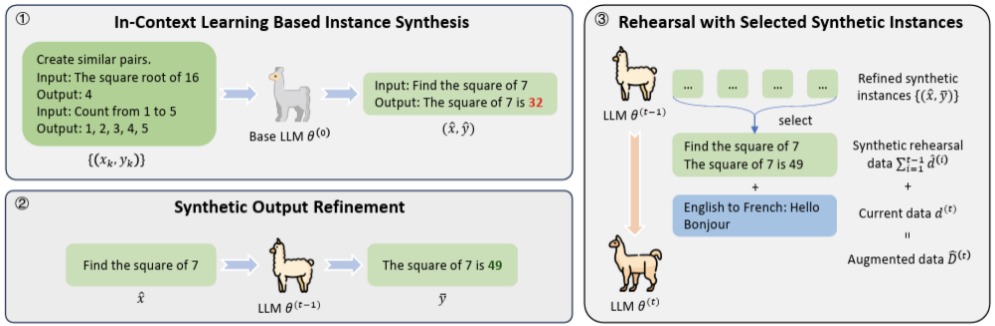
Llama-2-7B 128 tasks in 40 categories from SupNatInst

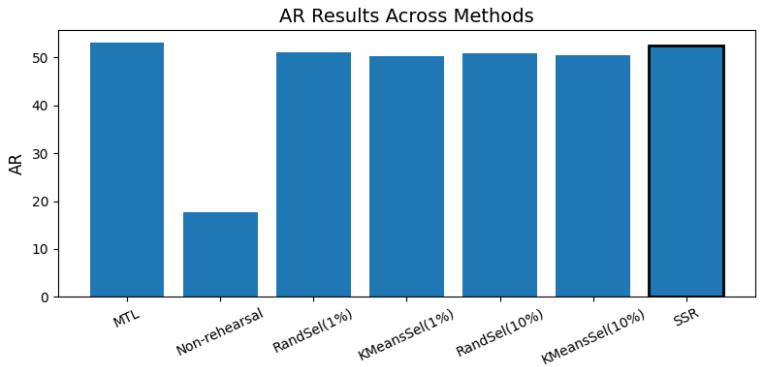
	Sup-NatInst-ST			
	Seen Tasks		Held-out Tasks	
Model	P-score	V-score	P-score	V-score
SFT	35.1	12.0	25.9	24.1
LoRA	33.7	12.4	26.7	23.0
L2	34.7	12.3	26.5	23.2
EWC	30.2	13.5	25.1	24.6
DARE	-	-	-	-
LM-Cocktail	-	-	-	-
PCLL	50.5	5.4	38.2	5.6
DCL	50.2	4.9	38.8	5.2
DYNAINST	50.9	4.6	38.7	4.4
InsCL	52.5	4.0	38.4	5.5
KPIG	52.2	3.5	42.5	1.7
INIT	43.2	5.3	∗43.8	∗1.5
MULTI	∗59.8	∗2.2	41.4	4.2

Y. He, X. Huang, M. Tang, L. Meng, X. Li, W. Lin, W. Zhang, and Y. Gao. Don't half-listen: Capturing key-part information in continual instruction tuning, ACL 2024.

CIT : Self-Synthesized Rehearsal

- Conventional rehearsal-based methods rely on previous training data to retain the model's ability, which may not be feasible in real-world applications, for instance CL on publicly released LLM checkpoint
- Self-Synthesized Rehearsal (SSR) that uses the LLM to generate synthetic instances for rehearsal.
  - the base LLM to generate synthetic instances, conducting in-context learning (ICL) with few-shot demonstrations (The ICL ability of LLMs tends to exhibit a significant degradation after supervised fine-tuning (SFT) on specific tasks)
  - latest LLM is used to refine the outputs of synthetic instances (synthetic instance retains the knowledge acquired by the latest LLM.)
  - select diverse high-quality synthetic instances for rehearsal (through clustering)
  - Llama-2-7b and SuperNI dataset (Wang et al., 2022)

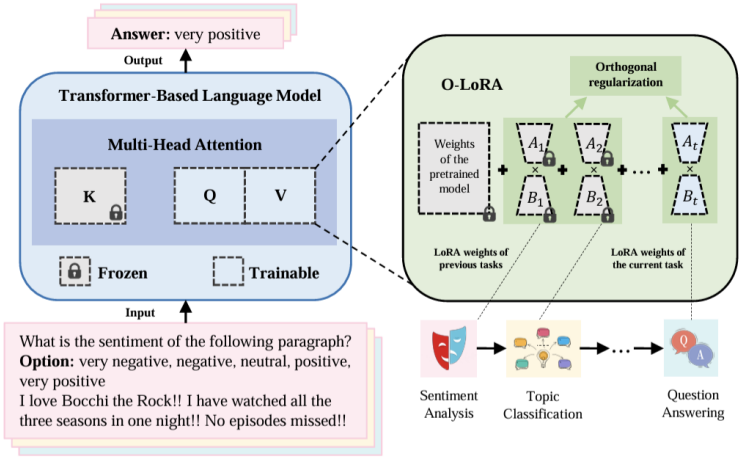




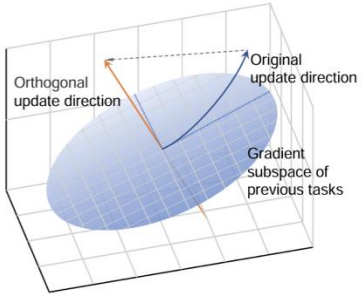
J.Huang,L.Cui,A.Wang,C.Yang,X.Liao,L.Song,J.Yao,andJ.Su. Mitigatingcatastrophicforgettinginlargelanguagemodelswithself-synthesized rehearsal, ACL 2024.

CFT : Orthogonal LORA

- Introduce O-LoRA, a simple and efficient approach for continual learning in language models, incrementally learning new tasks in orthogonal subspaces
- Orthogonal Gradient Descent (OGD) needs to store gradients of all previous data - intractable for large-scale language models, used PEFT based on LORA
- Hypothesis : LoRA parameters encapsulate crucial model update directions. Therefore, the gradient subspaces of previous tasks are succinctly represented by the LoRA parameters.
- Llama &B trained on – Alpaca data set - open-source multitask instruction tuning dataset introduced by Taori et al. (2023) Tasks from various domains, such as STEM, humanities, social sciences, and general knowledge.
- Tested on MMLU data set - Massive Multitask Language Understanding designed to evaluate the general knowledge and problem-solving



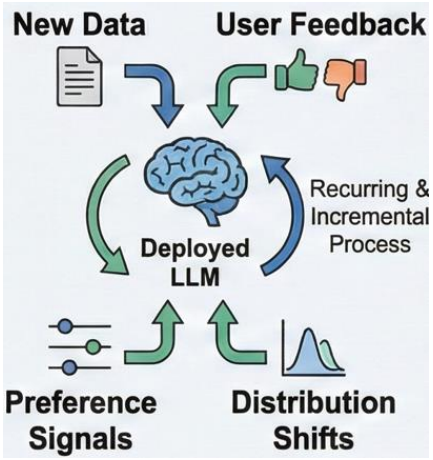
$$\min_{A_t,B_t} \underbrace{\mathcal{L}_t\left(f(x;W_0+B_tA_t^\top)\right)}_{\text{Task loss}} + \lambda \underbrace{\|U_{t-1}^\top A_t\|_F^2 + \|U_{t-1}^\top B_t\|_F^2}_{\text{Orthogonality constraint}}.$$



	MMLU
LLaMA-7B	34.4
Alpaca-LoRA	37.5
Alpaca-LoRA-CL	23.3
Alpaca-inc-LoRA-CL	28.6
Alpaca-OLoRA-CL	33.6

Continual Model Refinement (CMR): Post Deployment Improvement of LLMS

What Is CMR



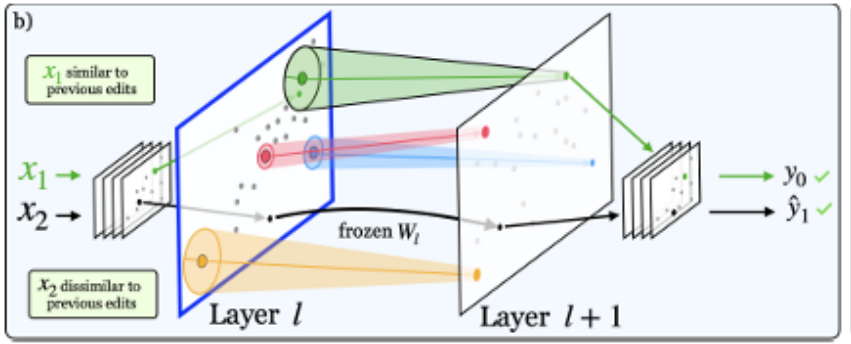
CMR is the recurring and incremental process of improving or correcting a deployed LLM.

- **Deployed language models decay over time due to shifting inputs, changing user needs, or emergent world-knowledge gaps. When such problems are identified, we want to make targeted edits while avoid modifying behaviors of pre-trained models.**
- **CMR allows safe, targeted, and continual improvement of a deployed LLM.** Modern methods use **model editing, retrieval-triggered updates and parameter isolation (e.g., LoRA experts)** to avoid broad interference and maintain alignment.

CMR : Lifelong Model Editing with Discrete Key-Value Adaptors

This method treats **hidden representations** of the language model as a *similarity key* to decide **when** to use the updated parameters for a new task.

GRACE enables **targeted, non-destructive model edits** by adding a lightweight **key–value memory** at selected layers, rather than modifying the model's original weights.



How it works

- Each edit stores: (h^{l-1},v)
  - a **key** (representing a latent activation pattern h^{l-1})
  - a **value** (the corrected or updated behavior v, learnt by fine tuning)
  - an **influence radius** (ε) controlling when the edit applies

Deferral decision at inference

- At each GRACE-enabled layer, the model:
  - Treats the current hidden activation as a **query**
  - Searches for the **closest stored key**
- If the query is **similar enough** (distance < ε): GRACE uses the value
- Else falls back to the original pretrained computation

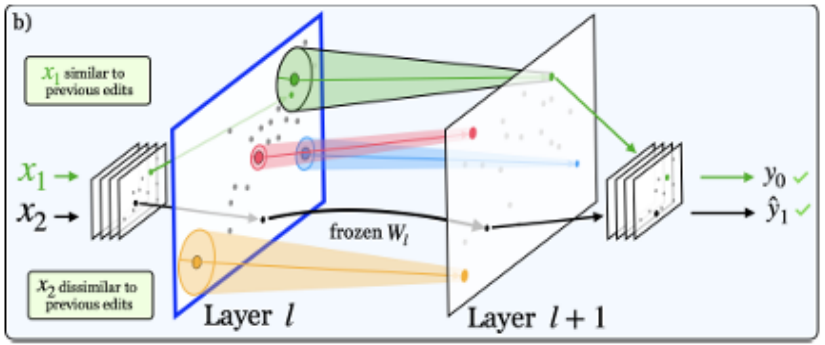
$$h^l = \begin{cases} \text{GRACE}(h^{l-1}) & \text{if } \min_i(d(h^{l-1}, \mathbb{K}_i)) < \epsilon_i \\ f^l(h^{l-1}) & \text{otherwise,} \end{cases}$$

T. Hartvigsen, S. Sankaranarayanan, H. Palangi, Y. Kim, and M. Ghassemi. Aging with grace:
Lifelong model editing with discrete key-value adaptors. In Advances in Neural Information Processing Systems, 2023.

CMR : Lifelong Model Editing with Discrete Key-Value Adaptors

This method treats **hidden representations** of the language model as a *similarity key* to decide **when** to use the updated parameters for a new task.

GRACE enables **targeted, non-destructive model edits** by adding a lightweight **key–value memory** at selected layers, rather than modifying the model's original weights.



- **Experiments to correct hallucination**: GPT-3 to generate 238 wikipedia-style biographies using subjects from WikiBio. Edits for all 238 biographies, creating 1392 sequential edits and 592 already-accurate outputs.
- **Test RetentionRate (TRR):** Edited model retains its performance on original testing data
- **Edit Retention Rate (ERR):** We check how well an edited model retains previous edits
- Metric used is perplexity - lower the better

Method	TRR	ERR	ARR
FT [25]	1449.3	28.14	107.76
FT+EWC [19]	1485.7	29.24	109.59
FT+Retrain [36]	2394.3	35.34	195.82
MEND [30]	1369.8	1754.9	2902.5
Defer [31]	8183.7	133.3	10.04
ROME [28]	30.28	103.82	14.02
Memory	25.47	79.30	10.07
GRACE	**15.84**	**7.14**	**10.00**

T. Hartvigsen, S. Sankaranarayanan, H. Palangi, Y. Kim, and M. Ghassemi. Aging with grace: Lifelong model editing with discrete key-value adaptors. In Advances in Neural Information Processing Systems, 2023.

CMR : A Dual-Memory Architecture for Lifelong Editing (WISE)

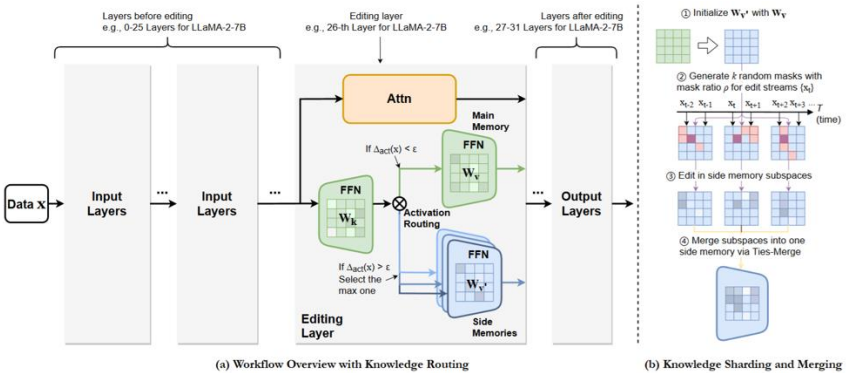
Existing methods for Continual refinement for LLMs fail to achieve reliability, Generalization and Locality. WISE resolves this by introducing a dual-memory system that isolates edits from the original model's knowledge.

Triangle in Lifelong Editing

Methods that edit model parameters (**long-term memory** e.g., ROME) suffer from poor locality, while retrieval based methods (**working memory,** e.g., GRACE) fail to generalize. This creates a fundamental trade-off.

- **Reliability:** Remembering all past edits across a long sequence.
- **Generalization:** Applying edits to paraphrased or semantically similar queries.
- **Locality:** Ensuring edits do not affect unrelated knowledge in the model.

The WISE Solution: Dual Memory, Routing, and Sharding



(a) Workflow Overview with Knowledge Routing (b) Knowledge Sharding and Merging

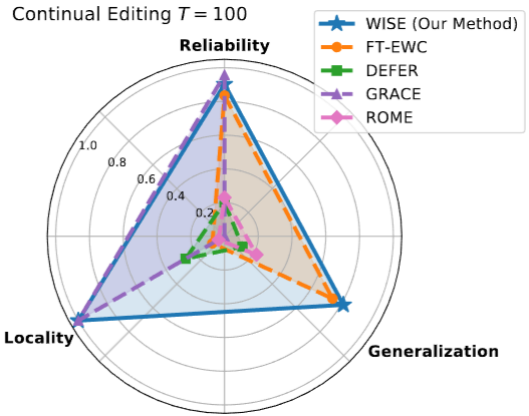
1. **Side Memory Creation:** A copy of a specific FFN layer's value matrix (w_v) is created to serve as a dedicated editable 'side memory' (W_v')
2. **Activation-Based Routing:** A router determines if an input x is related to an edit by calculating the activation of the side memory's corresponding FFN layer)
3. **Knowledge Sharding:** Incoming edits are distributed across distinct subspaces of the side memory defined by random masks. This prevents catastrophic forgetting.
4. **Conflict-Free Merging:** Periodically, the edited subspaces are merged into a single side memory using Ties-merge.

Wang, Peng, et al. "Wise: Rethinking the knowledge memory for lifelong model editing of large language models." Advances in Neural Information Processing Systems 37 (2024): 53764-53797.

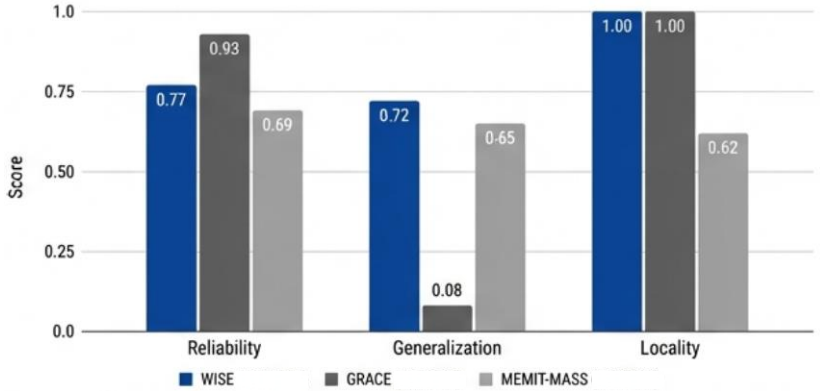
CMR : WISE Results

Across diverse tasks and thousands of sequential edits, WISE consistently outperforms state-of-the-art model editors by successfully balancing the three key metrics of Reliability, Generalization and Locality.

WISE Overcomes the Impossible Triangle
(ZsRE QA Task, LLaMA-2-7B)



Superiority in Long-Sequence Editing
(ZsRE QA Task, LLaMA-2-7B, T=1000 edits)



Key Takeaways

- **Breaks the Trade-Off: WISE** is the only method to maintain high performance across all three metric.
- **Outperforms SOTA by a Wide Margin:** The average score of **0.83** on the QA Task represents a significant jump over the next best method. In contrast GRACE's generalization score collapses to 0.08
- **Robust & Scalable:** This strong performance holds across models (LLaMA, Mistral, GPT-J), tasks (Question Answering, Hallucination Correction), and scales gracefully to **3,000+ edits**, demonstrating true lifelong learning capability.

Continual Model Alignment (CMA)

Why CMA is Needed

- A **single alignment stage** (e.g., RLHF, DPO) can **restrict** the model's behavior to a narrower distribution.
- Alignment updates often overwrite previously learned preferences; This phenomenon is called the "**Alignment Tax**"
- As factual knowledge, societal norms, and safety standards evolve, CMA ensures **ongoing alignment** with contemporary expectations.

Different CMA Approaches

RL Based CMA

- Standard RLHF/DPO/STeER modifies **many parameters at once**, causing large behavioral shifts.
- **Adaptive Model Averaging (AMA):** An alignment-aware continual adaptation mechanism that averages *multiple model layers* using adaptive weights to minimize alignment tax.
- **Continual Proximal Policy Optimization (CPPO):** Learns **example-level weighting** to decide when to strengthen policy alignment vs. retain earlier behaviors.

SL-Based CMA

- Frequent preference shifts require continual correction of supervised alignment datasets.
- **Continual Optimal Policy Fitting (COPF):** An adaptation of DPO that mitigates *sub-optimal policy fitting* and prevents over-optimization under continual updates.

CMA : Mitigating the Alignment Tax of RLHF

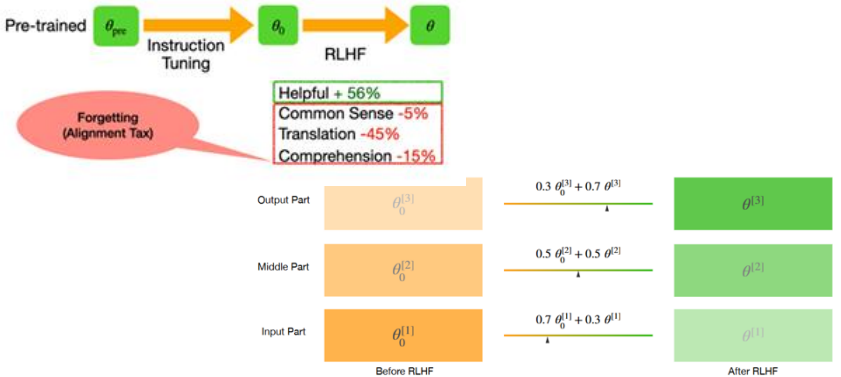
The Alignment-Forgetting Trade-off
Reinforcement Learning from Human Feedback (RLHF) aligns LLMs with human preferences, but degrades performance on the pre-trained abilities of a model (general language modelling – reasoning , coding, instruction following , generalization), known as the '**alignment tax**'

The Surprising Power of Model Averaging (MA)
A simple interpolation between the pre-RLHF and post-RLHF models (Model Averaging) achieved the strongest alignment-forgetting among all competing methods.

$$\theta_{avg} = \alpha\theta_{SFT} + (1-\alpha)\theta_{RLHF}$$

Key Insight: Layer-Specific Effects
The trade-off is not uniform across the model.
Critically, averaging lower-level layers (e.g., layers 1 – 8) produces a 'magical' improvement in both alignment reward and NLP task.

Tasks could share similar lower-level features, e.g., better word representation on low-level layers benefits both NLP and alignment tasks.



HMA adaptively interpolates between the pre and post RLHF models by assigning a unique averaging ration to each of the K transformer blocks.

$$\theta^{[k]}(K) := \alpha_k\theta^{[k]} + (1-\alpha_k)\theta_0^{[k]}, \forall k \in 1, ..., K.$$

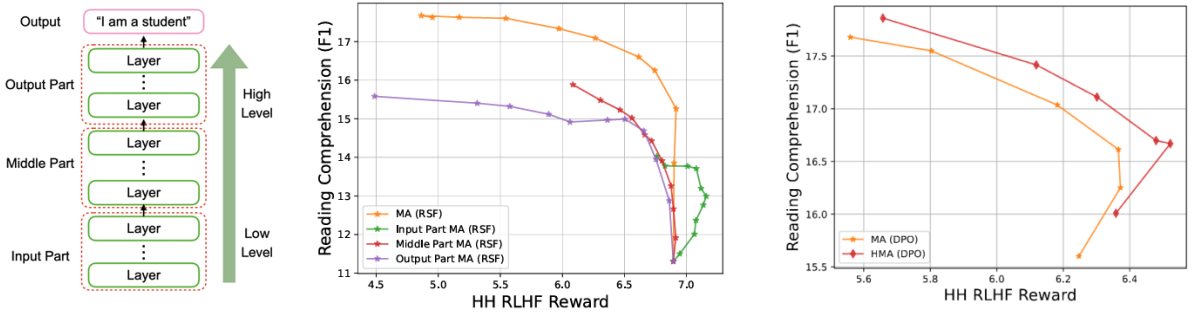
Optimization Goal
For a given level of 'forgetting' (proxied by the mean of the ratios,), HMA finds the optimal combination of layer wise ratios
That maximizes the alignment reward.

$$\max_{(\alpha_1,...,\alpha_K)\in\Omega} \mathbb{E}_x\mathbb{E}_{a\sim\pi_{\theta(K)}(\cdot|x)}\left[r^*(x,a)\right]. \quad \text{subject to } \frac{1}{K}\sum\alpha_k = \alpha$$

Lin, Yong, et al. "Mitigating the alignment tax of rlhf." Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing. 2024.

CMA : Mitigating the Alignment Tax of RLHF: Results

- HMA consistently outperformed vanilla MA strategies and it's performance was consistent across different model sizes and RLHF algorithms such as RSF,PPO and DPO.
- Alignment is evaluated using the "helpfulness and harmlessness dataset" (Bai et al., 2022) and average reward score across all prompts is reported as the HH RLHF.
- Alignment tax is evaluated on Commonsense QA, Reading comprehension, and Translation



Generalization to state of the art models on Alpaca benchmark

Model	Win-Rate	Reading	CommonSense	Trans
Zephyr-7B-β	8.10%	37.47	66.34	36.55
HMA (Ours)	**9.32%**	**38.93**	**66.55**	**37.23**
Zephyr-7B-Gemma	11.3%	41.15	66.3	38.09
HMA (Ours)	**11.5%**	**42.45**	**66.4**	**38.71**

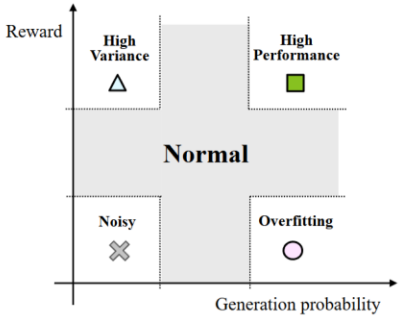
Conclusion: By leveraging the insight of layer-specific features sharing, HMA provides a more effective, principled, and broadly applicable method for mitigating the alignment tax in LLMs.

CMA : A Sample-wise Weighting Strategy for Continual Alignment (CPPO)

The Core Idea

- **The Insight**: Categorize Samples by Reward & Probability
  The foundation of CPPO is to classify each generated sample (x) into one of five types based on it's reward score R(x) and the policy's generation probability P(x). This enables a targeted, rather than uniform, approach to policy updates.

- **The Mechanism: Adaptive Sample-wise Weighting**
  The paper introduces two weights to modulate the PPO objective for each sample:
  - $\alpha(x)$ **Policy Learning Weight**: Controls the intensity of learning from a given sample
  - $\beta(x)$ **Knowledge Retention Weight**: Controls the penalty for deviating from the old policy for that sample.
- **Weighting Strategy**:
  - High-performance ($\alpha\uparrow, \beta\uparrow$):Consolidate knowledge for this sample.
  - High variance/ overfitting sample ($\alpha\uparrow, \beta\downarrow$): Learn new knowledge from this sample and force new sample to be different.
  - Noisy ($\alpha\downarrow, \beta\downarrow$): Decrease it's impact on learning.
  - Normal sample: Make no change to the PPO Objective.



The Final Objective Function:
This adaptive weighting strategy is integrated directly into a modified PPO loss function, creating a more stable and intelligent learning objective.

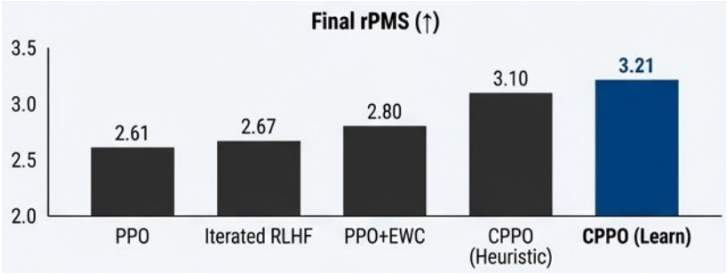
$$\mathbf{J}(\theta) = L_i^{\alpha\cdot CLIP+\beta\cdot KR+VF}(\theta)$$
$$= \mathbb{E}_i[\alpha(x)L_i^{CLIP}(\theta) - \beta(x)L_i^{KR}(\theta) - c\cdot L_i^{VF}(\theta)]$$

$$L_i^{KR}(\theta) = (\log P_{\pi_\theta}(x_i) - \log P_{\pi_{t-1}}(x_i))^2$$

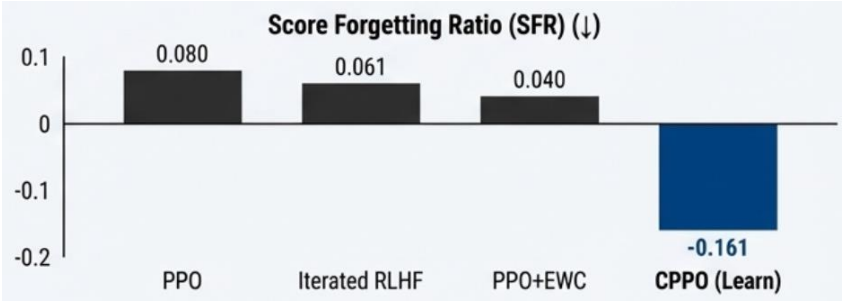
Zhang, Han, et al. "Cppo: Continual learning for reinforcement learning with human feedback." The Twelfth International Conference on Learning Representations. 2024.

CMA : CPPO Results…

- Evaluate method using the Reddit (Völske et al., 2017) dataset for summarization. Used the human preference data provided by CarperAI5. 1.3B gpt2-xl as RM and PM
- **Outperforms Baselines in Final Alignment**
  After learning multiple tasks, CPPO achieves the highest reference Preference Model Score (rPMS), indicating superior alignment with human preferences.



- **Prevents Forgetting and Enables Backward Transfer**
  While all baselines exhibit catastrophic forgetting, CPPO is the only method to achieve a negative SFR. This indicates backward transfer.



prompt string · *lengths* 73 2.37k	chosen string · *lengths* 27 591	rejected string · *lengths* 24 591
SUBREDDIT: r/relationships TITLE: My [21/M] girlfriend [19/F] broke…	TL;DR: My girlfriend and I broke up after she went through my Facebook…	TL;DR: Girlfriend went through my Facebook account and found message…
SUBREDDIT: r/relationships TITLE: My [21/M] girlfriend [19/F] broke…	TL;DR: My Girlfriend of 15 months went through my Facebook messages…	TL;DR: My girlfriend found messages in my Facebook from a girl I liked…
SUBREDDIT: r/AskReddit TITLE: Dear Reddit: Have you ever HAD to be an…	TL;DR: liked a girl but recognized it wouldn't work out in the long…	TL;DR: I had to be an asshole for a moral/good cause.

CMA : Continual Optimal Policy Fitting

A Non-RL Method for Evolving LLM Alignment

DPO's Limitations

- **Static by Design**: DPO is great for one-shot alignment, but it is not designed for adapting to human evolving human preferences over sequential tasks.
- **Risk of Over-optimization:** DPO aggressively widens the gap between preferred and rejected samples

$$\Delta \log \pi_\theta = \log \pi_\theta(y^+|x) - \log \pi_\theta(y^-|x)$$

$$\mathcal{L}_{\text{DPO}}(\theta) = -\mathbb{E}_{(x,y^+,y^-)}\Big[\log \sigma\big(\beta(\Delta \log \pi_\theta - \Delta \log \pi_{\text{ref}})\big)\Big].$$

The COPF Process

COPF introduces two continual learning specific approaches to cater to the task of continual alignment:

- **Replay Memory**: Maintaining a buffer of historical task data to merge with new task data.
- **Function Regularization**: When learning a new task, COPF employs a regularization loss to ensure the new policy does not differ significantly from optimal policies of the previous tasks.

DPO suffers from instability during the initial training stage, to prevent this COPF decouples the reward estimation from the estimation from the immediate policy state by explicitly determining rewards before fitting.

$$\mathcal{L}_{\text{COPF}} = -\mathbb{E}\left[\log \sigma\left(\beta\left(\Delta \log \pi_\theta - \Delta \log \pi_{\text{ref}} - \underbrace{\gamma \cdot C_t}_{\text{continual correction}}\right)\right)\right].$$

Zhang, Han, et al. "Copf: Continual learning human preference through optimal policy fitting." ACL 2025

CMA : COPF Results

Task Incremental Learning for Human Feedback (TIL-HF) benchmark.

	HH-RLHF	Reddit TL;DR	IMDB
Task	Q&A	Summarization	Text Continuation
Input	Question	Reddit POST	Partial Review
Output	Helpful & Harmless Answer	Summarized Reddit POST	Positive Sentiment Completion
Preference Metric	SteamSHP	GPT-j	DistilBERT
Train Set	35.2k	14.8k	24.9k
Valid Set	200	200	200
Test Set	1000	1000	1000

COPF achieves the highest overall accuracy while better mitigating catastrophic forgetting compared to strong baselines, including a continual version of DPO.

Method / Taxonomy	HH-RLHF SteamSHP (↑)	Reddit TL:DR Gpt-J (↑)	IMDB DistillBert (↑)	Overall performance AA (↑)	Overall performance AIA (↑)	Memory stability BWT (↑)	Memory stability FM (↓)
L2-Reg / function regularization	0.797	0.812	0.812	0.807	0.807	-0.028	0.031
AGM [29] / gradient projection	0.812	0.827	0.832	0.824	0.829	-0.022	0.024
EWC [20] / weight regularization	0.803	0.821	0.826	0.817	0.821	-0.026	0.026
MAS [21] / weight regularization	0.807	0.819	0.812	0.813	0.819	-0.027	0.027
LwF [22] / function regularization	0.813	0.841	0.833	0.829	0.820	-0.024	0.024
TFCL [23] / weight regularization	0.813	0.832	0.829	0.825	0.821	-0.021	0.021
DER++ [24] / experience replay	0.815	0.837	0.841	0.831	0.836	-0.017	0.019
DPO^C [4] / function regularization	0.825	0.875	0.844	0.848	0.862	-0.026	0.026
COPF^L / function regularization	0.837	0.877	**0.878**	**0.864**	0.851	**0.007**	**-0.007**
COPF^G / function regularization	**0.856**	**0.895**	0.815	0.855	**0.878**	-0.021	0.021

Benchmarks Datasets : TRACE : A Comprehensive Benchmark for Continual Learning in Large Language Models

- Existing continual learning benchmarks are typically too simple for modern aligned LLMs and do not measure **general abilities**, **instruction-following quality**, or **safety behavior** after incremental updates.
- TRACE defines a *rigorous, standardized continual learning benchmark*
- TRACE consists of **eight sequential tasks** spanning multiple domains and difficulty types. The tasks are chosen to *challenge* models across reasoning, domain knowledge, multilingual skills, code generation, and math reasoning.

Domain-specific

- **ScienceQA** — Science question answering
- **FOMC** — Classify the tone of a central-bank statement: *dovish* (pro-lower rates), *hawkish* (pro-higher rates), or *neutral*
- **MeetingBank** — Meeting transcript summarization

Multi-lingual

- **C-STANCE** — Chinese stance detection
- **20Minuten** — German text simplification

Code completion

- **Py150** — Python next-line completion

Mathematical reasoning

- **NumGLUE-calculus and mathematics** — Math reasoning (cm)
- **NumGLUE-data structures** — Math reasoning (ds)

Metric (name)	Brief explanation	Equation
Overall Performance		
(OP)	Overall performance at step t on all tasks learned so far (average over tasks 1..t).	$\mathrm{OP}_t = \frac{1}{t}\sum_{i=1}^{t} R^D_{t,i}$
Backward Transfer		
(BWT)	Backward transfer at step t (how learning new tasks affects old-task performance; negative means forgetting).	$\mathrm{BWT}_t = \frac{1}{t}\sum_{i=1}^{t-1}\left(R^D_{t,i} - R^D_{i,i}\right)$

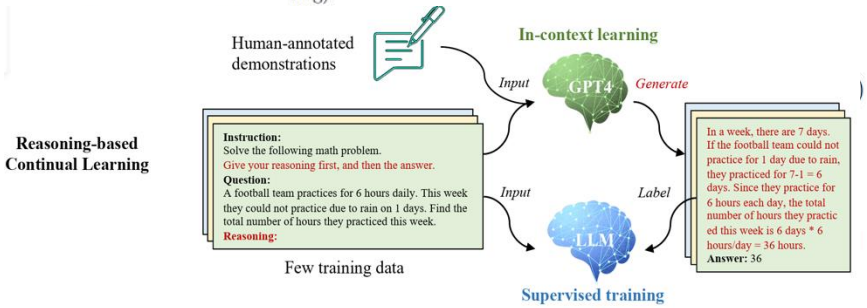


Table 1: OP(BWT) for all the baseline models and 3 baseline methods.

	ICL	SeqFT	LoraSeqFT	Replay
LLaMA-2-7B-Chat	38.9	48.7(−8.3%)	12.7(−45.7%)	55.5(2.6%)
LLaMA-2-13B-Chat	41.9	49.9(−7.0%)	28.0(−36.5%)	56.6(0.4%)
Vicuna-7B-V1.5	42.2	49.2(−8.4%)	33.4(−23.7%)	55.3(0.2%)
Vicuna-13B-V1.5	46.9	51.7(−5.9%)	31.6(−28.4%)	56.9(0.6%)
Baichuan2-7B-Instruct	44.6	43.4(−15.4%)	43.8(−9.0%)	51.7(1.1%)

Benchmarks Datasets : TemporalWiki-Benchmark for CPT

- TemporalWiki is a lifelong benchmark built from monthly Wikipedia/Wikidata snapshots to train and evaluate "ever-evolving" language models.
- It targets temporal misalignment (models trained on older data producing outdated facts).
- TWiki-Diffsets: training data created by taking the diff between consecutive monthly Wikipedia snapshots (new articles + changed/new sentences).
- Probes are labeled UNCHANGED vs CHANGED to evaluate stability (retain) and plasticity (update).
- Diff-based updating is reported as ~12× less compute than retraining on full snapshots (in their setup).

	# of Articles	# of Tokens
WIKIPEDIA-08	6.3M	4.6B
TWIKI-DIFFSET-0809	306.4K	347.29M
WIKIPEDIA-09	6.3M	4.6B
TWIKI-DIFFSET-0910	299.2K	347.96M
WIKIPEDIA-10	6.3M	4.7B
TWIKI-DIFFSET-1011	301.1K	346.45M
WIKIPEDIA-11	6.3M	4.6B
TWIKI-DIFFSET-1112	328.9K	376.09M
WIKIPEDIA-12	6.3M	4.7B

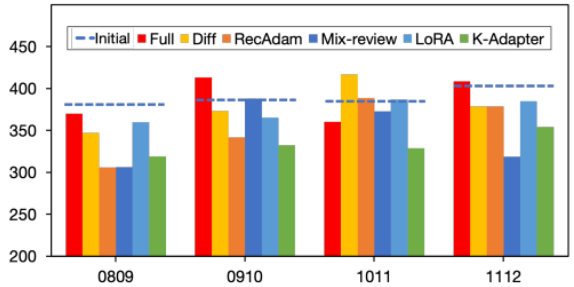


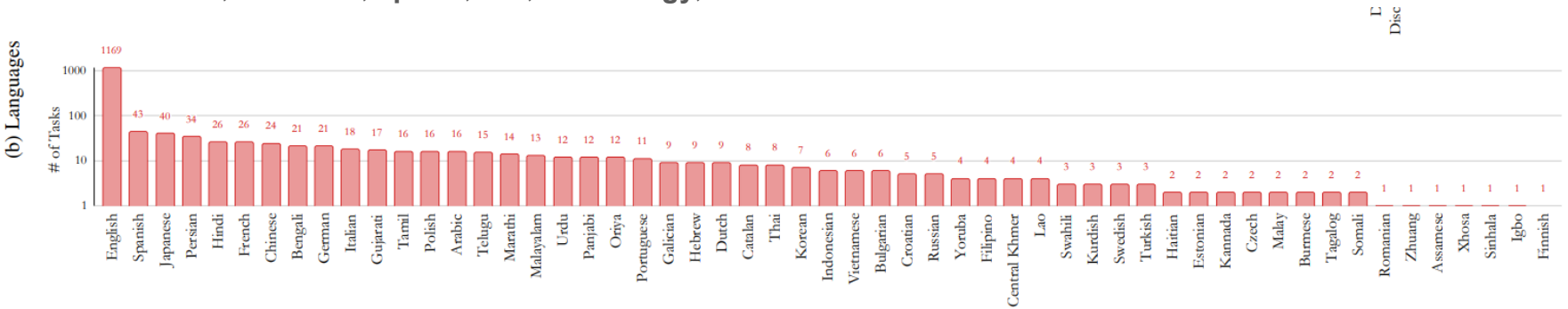
Figure 3: Average overall perplexity of TWIKI-PROBES. We average the perplexities of UNCHANGED and CHANGED with equal importance placed on stability and plasticity. The x-axis depicts the two-month intervals. A lower score indicates better performance.

Jang et al. (EMNLP 2022): "TemporalWiki: A Lifelong Benchmark

Benchmarks Datasets : SUPER-NATURAL INSTRUCTIONS (Super-NI)

- **Dataset Size**: Contains **1,616 NLP tasks** across **76 task types**, spanning **55 languages** and **33 domains**.
- **Task Types**: Includes a wide range of NLP tasks such as **classification**, **text generation**, **question answering**, **coreference resolution**, **paraphrasing**, and more.
- **Task Instructions**: Each task is paired with **detailed instructions**, including a **task definition**, **positive examples**, and **negative examples**.
- **Languages & Domains**: Covers tasks in **English and non-English languages** (576 non-English tasks) across diverse domains like **medicine**, **business**, **sports**, **law**, **technology**, and more.

statistic	
# of tasks	1616
# of task types	76
# of languages	55
# of domains	33
# of non-English tasks	576
avg. definition length (words per task)	56.6
avg. # of positive examples (per task)	2.8
avg. # of negative examples (per task)	2.4
avg. # of instances (per task)	3106.0

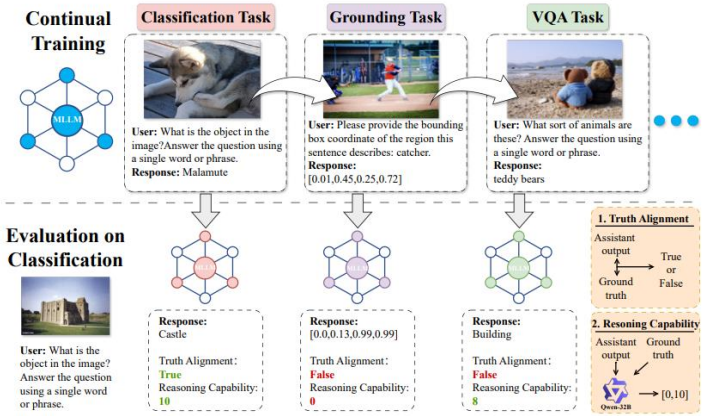


Benchmarks Datasets : CoIN (Continual Instruction Tuning for MLLMs)

- **Goal**: Benchmark continual instruction tuning and measure forgetting vs. retained ability
- **Coverage**: 10 datasets spanning 8 multimodal task categories
- Training setup: Sequential instruction tuning using LoRA (Low-Rank Adaptation)
- **Evaluation** (two views):

Instruction-Follow Accuracy: Standard task score (e.g., accuracy / exact match)

Reasoning / Knowledge Retention: LLM-graded score (focuses on reasoning quality)



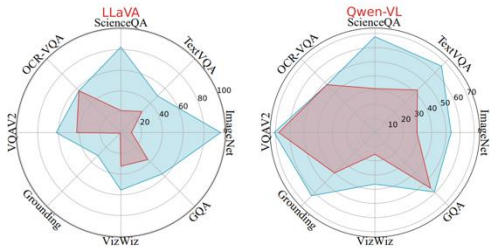
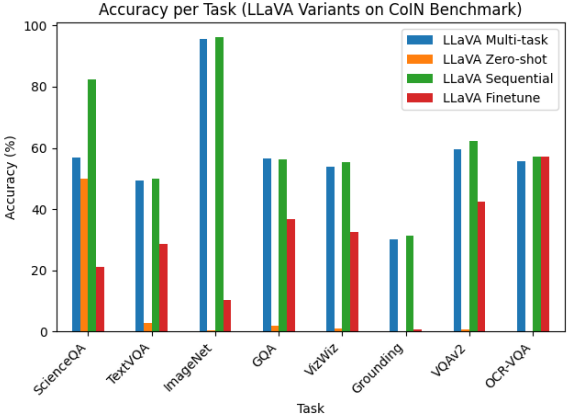


Table 1: The statistic of collected datasets and instructions in CoIN benchmark.

Task	Dataset	Instruction	Train Number	Test Number
Grounding	RefCOCO RefCOCO+ RefCOCOg	Please provide the bounding box coordinate of the region this sentence describes: <description>	55k	31k
Classification	ImageNet	What is the object in the image? Answer the question using a single word or phrase	129k	5k
Image Question Answering (IQA)	VQAv2	Answer the question using a single word or phrase	82k	107k
Knowledge Grounded IQA	ScienceQA	Answer with the option's letter from the given choices directly	12k	4k
Reading Comprehension IQA	TextVQA	Answer the question using a single word or phrase	34k	5k
Visual Reasoning IQA	GQA	Answer the question using a single word or phrase	72k	1k
Blind People IQA	VizWiz	Answer the question using a single word or phrase	20k	8k
OCR IQA	OCR-VQA	Answer the question using a single word or phrase	165k	100k



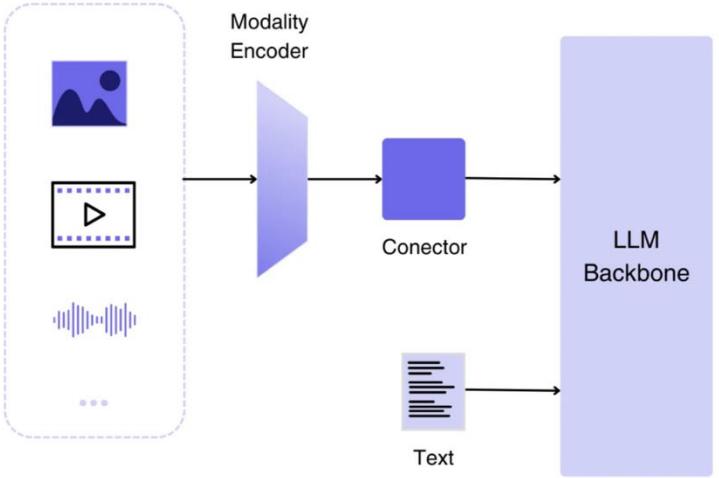
Continual Multimodal LLMs

Why Forgetting Happens in MLLMs

- Catastrophic forgetting in multimodal models often stems from **imbalanced or misaligned embedding representations**, not just sequential training.
- **Multimodal LLMs combine representations from fundamentally different spaces**: Text embeddings, Vision embeddings. These are heterogeneous and were not jointly pre-trained to lie on the same manifold.
- Each multimodal task requires different **alignments** between image/text tasks
- MLLMs depend heavily on ***modality bridges*** (adapters, projection layers). These bridges are: Not stable across tasks, gets updated aggressively, sensitive to downstream distribution shifts
- Different modalities are emerging throughout the incremental learning

Core Design Principles for Better CMLLMs

- Fixing or regularizing the **structure of the embedding space/**
- Allocate **task-aligned embedding directions** more evenly across the vector space.
- Use **representation regularization** to prevent dominant-dimension collapse.
- Encourage **modality separation or orthogonality** to reduce interference.



Retrieval augmented generation (RAG) and CL

- RAG augments an LLM with an **external, updatable memory** (documents, passages, databases) retrieved at inference time
- RAG can be seen as an external-memory-based alternative to continual learning : it updates **external knowledge**, not parameters, while CL updates model parameters
    - Avoids Catastrophic Forgetting Entirely : No parameter overwriting, New knowledge added by indexing new documents, Zero Replay Cost - No need to store old training data, Handles Temporal & Factual Drift
- Parameter-based continual learning performs updates via gradient descent, whereas retrieval-augmented generation updates knowledge by modifying the underlying corpus
- RAG often **outperforms CPT for factual freshness**
- RAG **does NOT solve**:
    - Continual instruction tuning, Skill acquisition (reasoning, coding), Behavioral alignment, Multimodal grounding

Lewis et al., 2020. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks, NeurIPS 2020

CL in LLMs : Future Directions

- Continual learning community tends to prioritize empirical research over theoretical exploration.
  - Learning Dynamics in Continual Pre-Training for Large Language Models [1]
  - Unified Domain Incremental Learning (UDIL) in [2] proposes upper bounds for intra-domain and cross-domain distillation losses.
  - Applying these existing theories directly to continual LLMs can be imprudent, given their pre-trained, large-scale nature and loss functions.
- More sophisticated and accurate data mixing strategies and efficient replay sample selection mechanisms are needed
- The long-term memory inherent in the whole set of parameters of LLMs often lacks interpretability and explicit manipulability
- The existing works on mostly focused on domain incremental and task incremental learning settings, needs exploration in class incremental learning for LLMs.

[1]Wang, X., Tissue, H., Wang, L., Li, L., & Zeng, D. D. (2025). *Learning dynamics in continual pre-training for large language models*. arXiv preprint arXiv:2505.07796.[
[2] Shi, H., & Wang, H. (2023). *A unified approach to domain incremental learning with memory: Theory and algorithm.* In H. Larochelle et al. (Eds.), Advances in Neural Information Processing Systems, 36. NeurIPS.

Key Python Libraries for Continual Learning

- **Avalanche (PyTorch):** End-to-end continual learning library with benchmarks, many implemented baselines/SOTA algorithms, experiment utilities, and reproducible pipelines. Great starting point for deep-learning CL research.
- **Continuum:** Dataset and scenario utilities for setting up class-incremental and task-incremental experiments easily (good for data handling & scenario generation).
- **PyCIL (Python CIL):** Focused toolbox for class-incremental learning: implements classical CIL methods (iCaRL, LwF, EWC variants) and evaluation pipelines.
- **Renate:** Production-oriented library for automating continual retraining and model maintenance (built on PyTorch & PyTorch Lightning) — useful for applied systems and MLOps workflows.
- **SequeL:** Research-friendly framework (PyTorch + JAX support) designed for extensibility across regularization, replay, and architectural CL methods.
- **River (stream learning):** Lightweight streaming/online learning library focused on concept drift and incremental models (ideal for non-deep or resource-constrained streaming setups).
- **ContinualLM:** Github repository of an extensive Continual Learning framework for LLMs focused on continual DAPT containing pytorch implementation of multiple SOTA methods.

Avalanche

PyCIL
A Python Toolbox for
Class-Incremental Learning
Via Model Reuse

River

References

- Chen, L., Zaharia, M., and Zou, J., 2024. How is ChatGPT's behavior changing over time?. Harvard Data Science Review, 6(2).
- Gogoulou, E., Lesort, T., Boman, M., and Nivre, J., 2024. Continual learning under language shift.
- Li, C.-A., and Lee, H.-Y., 2024. Examining forgetting in continual pre-training of aligned large language models.
- Qin, Yujia, et al., 2023. "Recyclable tuning for continual pre-training." arXiv preprint arXiv:2305.08702.
- Gururangan, Suchin, et al., 2023. "Demix layers: Disentangling domains for modular language modeling." Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies.
- Chen, Wuyang, et al., 2023. "Lifelong language pretraining with distribution-specialized experts." International Conference on Machine Learning, PMLR.
- J. Jang, S. Ye, S. Yang, J. Shin, J. Han, G. Kim, S. J. Choi, and M. Seo. Towards continual knowledge learning of language models. In ICLR Jang, J., Ye, S., Lee, C., Yang, S., Shin, J., Han, J., Kim, G., and Seo, M., 2022. Temporalwiki: A lifelong benchmark for training and evaluating ever-evolving language models.
- Yan, Y., Xue, K., Shi, X., Ye, Q., Liu, J., and Ruan, T., 2023. "Af adapter: Continual pretraining for building Chinese biomedical language model." IEEE International Conference on Bioinformatics and Biomedicine (BIBM).
- Acikgoz, E. C., İnce, O. B., Bench, R., Boz, A. A., Kesen, İ., Erdem, A., Erdem, E., 2024. "Hippocrates: An open-source framework for advancing large language models in healthcare." arXiv preprint arXiv:2404.16621.
- Paul, I., Luo, J., Glavaš, G., and Gurevych, I., 2024. "Ircoder: Intermediate representations make language models robust multilingual code generators."
- Lin, Z., Deng, C., Zhou, L., Zhang, T., Xu, Y., Xu, Y., He, Z., Shi, Y., Dai, B., Song, Y., Zeng, B., Chen, Q., Shi, T., Huang, T., Xu, S., Wang, S., Fu, L., Zhang, W., He, J., Ma, C., Zhu, Y., Wang, X., Zhou, C., 2023. "Geogalactica: A scientific large language model in geoscience."
- Chen, J., Wang, X., Gao, A., Jiang, F., Chen, S., Zhang, H., Song, D., Xie, W., Kong, C., Li, J., Wan, H., Li, B., 2023. "Huatuogpt-ii, one-stage training for medical adaption of llms." CoRR, abs/2311.09774.
- Zhang, Y., Wang, X., and Yang, D., 2022. "Continual sequence generation with adaptive compositional modules." In S. Muresan, P. Nakov, and A. Villavicencio (Eds.), ACL 2022.
- Liang, Yan-Shuo, and Wu-Jun Li., 2025. "Gated Integration of Low-Rank Adaptation for Continual Learning of Language Models (GainLoRA)." NeurIPS 2025.

References

- P. Colombo, T. P. Pires, M. Boudiaf, D. Culver, R. Melo, C. Corro, A. F. T. Martins, F. Esposito, V. L. Raposo, S. Morgado, and M. Desa. Saullm-7b: A pioneering large language model for law, 2024. T. Computer. Redpajama: an open dataset for training large language models, 2023
- J. Chen, X. Wang, A. Gao, F. Jiang, S. Chen, H. Zhang, D. Song, W. Xie, C. Kong, J. Li, X. Wan, H. Li, and B. Wang. Huatuogpt-ii, one-stage training for medical adaption of llms. CoRR, abs/2311.09774, 2023 Z.
- Lin, C. Deng, L. Zhou, T. Zhang, Y. Xu, Y. Xu, Z. He, Y. Shi, B. Dai, Y. Song, B. Zeng, Q. Chen, T. Shi, T. Huang, Y. Xu, S. Wang, L. Fu, W. Zhang, J. He, C. Ma, Y. Zhu, X. Wang, and C. Zhou. Geogalactica: A scientific large language model in geoscience, 2023
- Q. Xie, Q. Chen, A. Chen, C. Peng, Y. Hu, F. Lin, X. Peng, J. Huang, J. Zhang, V. Keloth, et al. Me llama: Foundation large language models for medical applications. arXiv preprint arXiv:2402.12749, 2024
- He, Y., Huang, X., Tang, M., Meng, L., Li, X., Lin, W., Zhang, W., and Gao, Y., 2024. "Don't half-listen: Capturing key-part information in continual instruction tuning." ACL 2024.
- Huang, J., Cui, L., Wang, A., Yang, C., Liao, X., Song, J., Yao, J., and Su, J., 2024. "Mitigating catastrophic forgetting in large language models with self-synthesized rehearsal." ACL 2024.
- Zhang, Han, et al., 2024. "Cppo: Continual learning for reinforcement learning with human feedback." The Twelfth International Conference on Learning Representations.
- Zhang, Han, et al., 2025. "Copf: Continual learning human preference through optimal policy fitting." ACL 2025.
- Zhao, Shu, Zou, Xiaohan, Yu, Tan, and Xu, Huijuan., 2024. "Reconstruct before Query: Continual Missing Modality Learning with Decomposed Prompt Collaboration."
- He, J., Guo, H., Tang, M., and Wang, J., 2023. "Continual instruction tuning for large multimodal models."
- Wang, Peng, et al., 2024. "Wise: Rethinking the knowledge memory for lifelong model editing of large language models." Advances in Neural Information Processing Systems 37.
- Lin, Yong, et al., 2024. "Mitigating the alignment tax of rlhf." Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing.
- Shi, H., & Wang, H., 2023. "A unified approach to domain incremental learning with memory: Theory and algorithm." In Advances in Neural Information Processing Systems, 36.
- Lewis, P., et al., 2020. "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks." NeurIPS 2020.

Dr. Srijith P. K.
Bayesian Reasoning and Intelligence (BRAIN) Lab
Department of Computer Science and Engineering
Department of Artificial Intelligence (Affiliated)
Indian Institute of Technology Hyderabad
https://www.iith.ac.in/cse/srijith/
https://sites.google.com/view/brainiith/home
srijith@cse.iith.ac.in

Dupati Srikar Chandra
BRAIN
IIT Hyderabad

Sumanta Manna
BRAIN
IIT Hyderabad

Bayesian Reasoning And INtelligence