LexTempus: Enhancing Temporal Generalizability of Legal Language Models Through Dynamic Mixture of Experts

Santosh T.Y.S.S [1], **Tuan-Quang Vuong** [1,2]*
[1] School of Computation, Information, and Technology
Technical University of Munich, Germany
[2] Interdisciplinary Centre for Security, Reliability and Trust (SnT)
University of Luxembourg, Luxembourg
santosh.tokala@tum.de; quang.vuong@uni.lu

Abstract

The rapid evolution of legal concepts over time necessitates that legal language models adapt swiftly accounting for the temporal dynamics. However, prior works have largely neglected this crucial dimension, treating legal adaptation as a static problem rather than a continuous process. To address this gap, we pioneer LexTempus, a dynamic mixture of experts model that explicitly models the temporal evolution of legal language in a parameter-efficient online learning framework. LexTempus starts with a single lightweight adapter expert and dynamically expands by adding new experts as significant deviations in the data distribution are detected. This self-expansion strategy allows LexTempus to adapt to new information without forgetting past knowledge, thereby improving temporal generalization. We use a a non-parametric similarity-based router to merge relevant experts into a unified expert for each test instance, ensuring efficient inference without additional overhead. We validate the effectiveness of LexTempus on ECHR and EU case law datasets, demonstrating its superiority in both perplexity and open-ended text generation quality metrics.

1 Introduction

The integration of language models into the legal ecosystem marks a pivotal shift in how legal tasks are approached, ranging from drafting legal briefs to ensuring corporate compliance (Tiwari et al., 2024; Ziffer, 2023), with the potential to revolutionize the practice of law by aiding in understanding, analyzing, and generating legal documents (Frankenreiter and Nyarko, 2022; Santosh et al., 2025a). Traditional static models, trained on fixed corpora, may suffice in domains with stable knowledge, but legal systems are inherently dynamic, with new statutes, regulations, and case law constantly reshaping interpretative frameworks (Santosh et al., 2024c). For instance, shifts in legal frameworks occur through legislative changes like the GDPR and California Consumer Privacy Act, which have transformed data privacy practices; landmark rulings such as *Obergefell v. Hodges* and *Brown v. Board of Education*, which have reshaped same-sex marriage rights and racial segregation laws in the U.S.; regulatory updates from entities like the SEC and Basel Accords, influencing corporate and banking regulations; emerging issues such as autonomous vehicles and cryptocurrency, which prompt new legal standards; and international treaties like the Paris Agreement and the Convention on Cybercrime, impacting national laws on environmental and digital crimes. Additionally, evolving societal attitudes have driven the recognition of environmental rights and increased concerns over digital privacy, while external events like the COVID-19 have necessitated legal adaptations in telehealth.

This evolving nature of legal concepts presents a significant challenge for static models, which, if not regularly updated, risk becoming obsolete or providing inaccurate outputs based on outdated information (Dahl et al., 2024; Santosh et al., 2024a). To address this, language models need to continuously integrate and learn from new legal data to remain aligned with the most recent legal developments. Such adaptive models would ensure that legal practitioners have access to the most current and relevant information, thereby enhancing the reliability of AI-driven tools.

However, retraining solely on new data can lead to overfitting on recent information and catastrophic forgetting of previously learned data (McCloskey and Cohen, 1989; Nguyen et al., 2019), which undermines temporal generalizability (Yao et al., 2022; Lin et al., 2021; Lazaridou et al., 2021), a critical requirement in our dynamic and non-stationary legal domain. To address these chal-

* Work done during his study at the Technical University of Munich.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 6608–6624

lenges, the field of Continual Learning focuses on developing algorithms that balance knowledge expansion (plasticity) with knowledge retention (stability), enabling models to robustly extrapolate into the future in this temporal shift setting. Traditional CL strategies include experience replay (Lopez-Paz and Ranzato, 2017; Chaudhry et al., 2019; de Masson D'Autume et al., 2019), parameter or representation regularization (Kirkpatrick et al., 2017; Schwarz et al., 2018; Chen et al., 2020), and modular or isolated architectures (Yoon et al., 2017; Rusu et al., 2016; Qin et al., 2022b). These approaches typically involve training models from scratch with randomly initialized parameters, necessitating the retraining of all model parameters. However, with the growing trend of adapting pre-trained base models to new data, which already possess inherent generalization capabilities, a trade-off emerges: fully fine-tuning the weights to accommodate new data can erode the model's generalizability, while fixing the backbone prevents the integration of new information. Moreover, as models grow in size, fine-tuning all parameters becomes increasingly computationally expensive.

To address these limitations, recent approaches have adopted lightweight, parameter-efficient trainable modules that allow for the interpolation of generalization capabilities on newly learned data. While these methods have proven effective in class-, domain-, or task-incremental settings—where task boundaries are clearly defined and known in advance—they often rely on explicit boundary signals to switch between classes, domains, or tasks, and to train separate modules for each. However, this assumption does not hold in more complex scenarios of temporal adaptation, where data is continuous and boundaries are neither predefined nor explicitly supervised. Extending these approaches to boundary-unaware, online settings presents two key challenges: (i) determining when to add a new module during training, ensuring timely expansion without compromising past knowledge, and (ii) deciding which module to use during inference, given that no task signal is available.

To handle this boundary unaware, non-stationary temporal shift in online setting, we propose LexTempus, a dynamic mixture of experts model that initially starts with one lightweight adapter expert module and progressively introduces additional expert blocks as needed throughout the training process. This self-expansion strategy is triggered by statistically significant deviations in the loss function of the current expert, signaling the onset of data that cannot be effectively captured by existing experts due to a change in distribution, thus warranting the addition of a new expert to the pool. By continuously expanding the expert pool in response to these shifts, the model can better capture and adapt to the evolving distribution of data over time without forgetting past knowledge, thereby enhancing generalization. Furthermore, to comprehensively leverage the knowledge stored in different experts, we employ a non-parametric similarity-based router to estimate the relevance of each expert to a particular test instance. We then merge the different expert modules into a unified expert, inspired by model merging literature, based on these relevance probabilities. This approach tailors the model to the specific test instance, facilitating effective knowledge sharing across different experts and yielding more robust results. We demonstrate the effectiveness of our dynamic mixture of experts algorithm on both ECHR and EU case law documents, evaluating not only language modeling perplexity performance but also open-ended text generation metrics.

2 Related Work

Temporal Adaptation addresses the challenge of model performance deterioration over time due to naturally occurring distribution shifts (Schlimmer and Granger, 1986; Widmer and Kubat, 1993; Jaidka et al., 2018; Yao et al., 2022; Gorman and Bedrick, 2019). This can be caused due to (1) the dynamic nature of language (Rosin et al., 2022; Röttger and Pierrehumbert, 2021; Loureiro et al., 2022; Agarwal and Nenkova, 2022; Amba Hombaiah et al., 2021; Rijhwani and Preoţiuc-Pietro, 2020; Luu et al., 2022; Jaidka et al., 2018) and (2) the update of factual information (Margatina et al., 2023; Jang et al., 2021, 2022; Lazaridou et al., 2021; Dhingra et al., 2022; Liska et al., 2022).Temporal generalization has been explored both in upstream Language Model pre-training (Lazaridou et al., 2021; Loureiro et al., 2022; Jang et al., 2021, 2022; Dhingra et al., 2022; Jin et al., 2022; Amba Hombaiah et al., 2021) and in downstream tasks, such as sentiment analysis (Lukes and Søgaard, 2018; Agarwal and Nenkova, 2022; Guo et al., 2023b), named entity recognition (Rijhwani and Preoţiuc-Pietro, 2020; Onoe et al., 2022), question answering (Liska et al., 2022; Shang et al., 2022), headline generation (Søgaard et al., 2021), rumor detection

(Mu et al., 2023; Hu et al., 2023), spoken language understanding (Gaspers et al., 2022). model explainability (Zhao et al., 2022), document classification (Röttger and Pierrehumbert, 2021; Chalkidis and Søgaard, 2022; Huang and Paul, 2018; Santosh et al., 2024c), abusive language detection (Jin et al., 2023; Florio et al., 2020), topic modeling (Zhang et al., 2023b) and readmission prediction in the health care context (Guo et al., 2022, 2023a).

In the legal domain, Chalkidis and Søgaard (2022) and Santosh et al. (2024c) have examined temporal generalization in downstream multi-label legal classification tasks. Chalkidis and Søgaard (2022) suggested that temporal drift arises from shifts in label distribution over time, whereas Santosh et al. (2024c) emphasized that even label-specific vocabulary undergoes temporal changes. The latter proposed an incremental training framework that respects the temporal order of data, in contrast to previous approaches that treated the entire training dataset as a homogeneous entity, ignoring temporal shifts in input text distribution.

Our work extends this investigation to the upstream pre-training of language models on legal corpora in an "online" fashion, addressing the evolving legal context due to changes in interpretation, legislation, and precedents—a topic that, to the best of our knowledge, has not been previously explored. We focus on a challenging online setting, simulating real-world deployment scenarios where the model is exposed to a continuous stream of incrementally available legal information. The model must adaptively learn over time, integrating new legal developments while retaining previously acquired knowledge.

Related work on Continual Learning & Model Merging can be found in App. A.

3 Task & Datasets

We describe the task of online learning (Hazan et al., 2016; Shalev-Shwartz et al., 2012) over a distribution-varying stream S revealing data sequentially over steps $t \in \{1, 2, \ldots, \infty\}$. At each step t, the stream reveals data x_t from S. The model makes predictions for x_t using the current model m_{t-1}. The stream reveals the true labels and then the learner updates the model m_t using a fixed budget of computation and memory. The task of language modeling is to estimate the probability of next token given the context sequence of tokens.

We design our experiments to explore the online adaptation of language models using the following two legal judgments corpora from different jurisdictions. These legal corpora inherently exhibit distributional varying property of stream due to the dynamic nature of the legal domain (Sec. 5.1)—a result of the judiciary's role in interpreting ambiguous and vague legal formulations to settle open questions through landmark cases, which then influence subsequent legal discourse (Santosh et al., 2024a). This evolving legal landscape makes it an ideal testbed for studying online language modeling. In our setup, the data stream is naturally ordered by the timestamps of the documents. At each timestamp, the stream reveals an entire document, which is then segmented into multiple data instances to accommodate the context length of the language model. Under this online setup, models will have their parameters updated before the stream reveals document of the next step.

EU CaseLaw contains judgements of the Court of Justice of the European Union that are accessible via the EUR-Lex platform. We obtain this documents collection of Multi-EURLEX (Chalkidis et al., 2021) and crawl EUR-Lex platex for the date of the document. We filter the English portion of this dataset, consisting of 29,856 documents spanning from 12 February 1952 to 9 June 2022.

ECHR CaseLaw comprises of case judgements heard by the European Court of Human Rights and is publicly accessible via HUDOC, the official court database. These cases involve the adjudication of complaints by individuals against states for alleged violations of their rights, as enshrined in the European Convention of Human Rights. We obtain the most recent cleaned version of the dataset from Santosh et al. (2024b) which consists of 15,729 cases in English language from 14 November 1960 to 28 July 2022.

4 LexTempus: Our Method

In this section, we introduce our Dynamic Mixture of Experts (DMoE) algorithm which keep pre-trained model frozen and only add trainable adapter modules as experts. Our approach automatically determine addition of new expert into the growing pool of experts on demand for handling automatically detected novel patterns and dynamically aggregates experts tailored for each test instance. This is achieved through the incorporation of two crucial components: (1) Clustering based non-parametric router for aggregation of experts: This router deter-

mines the relevance of each expert to the current test instance and use those probability weights to merge these expert parameters into an unified expert. This enables different instances to activate varying numbers of experts. (2) A self-expansion strategy based on loss values to decide when to add new expert to the pool of experts. Overall process is illustrated in Algorithm 1.

Algorithm 1 Dynamic Mixture of Experts

Input: Pre-trained model M_{frozen}, stream of data $\mathcal{S} = \{x_1, x_2, \dots\}$, initial expert pool $\mathcal{E} = \{\mathbf{e}_1\}$, clustering algorithm (DBSCAN), expansion threshold τ
Output: Predictions $\hat{y}_t$ for each test instance x_t
for each time step t **do**
    Step 1: Embedding Generation
    Compute the semantic embedding $\mathbf{z}_t = M_{\text{frozen}}(x_t)$
    Step 2: Expert Aggregation
    for each expert $\mathbf{e}_i \in \mathcal{E}$ **do**
        Retrieve cluster centroids $\mathcal{C}_i = \{c_{i1}, c_{i2}, \dots, c_{ik}\}$ from DBSCAN on embeddings used to train $\mathbf{e}_i$
        Compute distances $d_i = \min_j \|\mathbf{z}_t - c_{ij}\|$ for $j = 1, \dots, k$
    end for
    Normalize distances: $w_i = \frac{1/d_i}{\sum_{j=1}^{|\mathcal{E}|} 1/d_j}$ for each expert $\mathbf{e}_i$
    Aggregate experts: $\mathbf{e}_{\text{merged}} = \sum_{i=1}^{|\mathcal{E}|} w_i \mathbf{e}_i$
    Step 3: Prediction
    Predict $\hat{y}_t = M_{frozen} \circ \mathbf{e}_{\text{merged}}(x_t)$
    Step 4: Self-Expansion Check
    Compute perplexity $P_t = \text{Perplexity}(\mathbf{e}_{\text{merged}}, x_t)$
    if $P_t > \text{z-score}(P_{t-\text{window}:t}) + \tau$ **then**
        Add new expert: $\mathbf{e}_{\text{new}} \leftarrow$ Initialize with weights from previous expert
        $\mathcal{E} \leftarrow \mathcal{E} \cup \{\mathbf{e}_{\text{new}}\}$
    end if
    Step 5: Expert Training
    Train only the current expert $\mathbf{e}_{\text{current}}$ with new data (x_t, y_t)
end for

4.1 Dynamic Aggregation of experts

We start by obtaining semantic embeddings from the frozen pre-trained model for the data that was used to train each expert. These embeddings are then clustered using a density-based clustering algorithm, DBSCAN (Ester et al., 1996), to identify the nearest cluster centroids, which serve as prototypes for each expert, capturing the essential semantics of various inputs captured by each expert. Rather than relying on a single prototype, we employ multiple prototypes for each expert, inspired by Tyss et al. (2024). This approach is motivated by the observation that the training data for each expert can exhibit significant variability, leading to diverse contextual embeddings distributed across the embedding space. Averaging these embeddings into a single prototype could dilute their specificity, so multiple prototypes per expert are utilized to more effectively capture the diverse instances within each expert's data.

During inference, when a new test instance is presented, we compute the embedding for this instance using the same frozen pre-trained model and measure the distance between this instance's embedding and the centroids corresponding to each expert. The minimum distance among all the centroids corresponding to a particular expert is identified for each expert and the inverse of these distances are normalized across different experts to produce relevance probabilities. These probabilities indicate the degree to which each expert is pertinent to the current test instance. Finally, based on these relevance probabilities, the parameters of the relevant experts are merged to create a unified expert, inspired by works from model merging (Wortsman et al., 2022; Chronopoulou et al., 2023; Ainsworth et al., 2022; Tam et al., 2024), which is then used to make predictions for the test instance. Instead of merging the output predictions similar to ensemble, model merging directly enables element-wise merging of all experts in their parameter space. to create an into unified expert, reducing the inference cost and enhancing generalization capabilites.

This approach dynamically adjusts the contribution of each expert based on the specific characteristics of the test instance data, resulting in more accurate and contextually appropriate predictions. In the context of long legal documents, which may need to be split into multiple input instances due to context length limitations of langauge model, this method further allows for the activation of different experts for each instance within a document, accounting for the varying arguments and distinct allegations present in each document.

4.2 Self-Expansion Strategy

During training, the addition of new experts to the pool is crucial when a novel pattern in the data stream indicates a shift or drift from prior data. This is identified through a statistically significant deviation in the perplexity (loss) value of the current expert, beyond a predefined expansion threshold. Such a deviation signals that the current expert may struggle to accommodate the new data without sacrificing previously learned information due to its capacity limitations. To detect this expansion signal, we maintain a sliding window or moving average of perplexity values.

When an expansion signal is triggered, a new adapter module is inserted into the transformer block and since then, training of the new expert begins. During this process, only the current active expert is trained, while all previously learned experts remain fixed. This strategy ensures that each expert focuses on a specific subset of data, which is beneficial for retaining past performance and enhancing generalization.

Additionally, whenever a new expert is added to the pool, it is initialized with the weights of the last adapter trained in the preceding time step. This helps the new expert inherit previously acquired knowledge, particularly useful when there is a limited number of training instances for the new expert, ensuring it can generalize effectively to the evolving data distribution.

5 Experiments

We use GPT-2 (Radford et al.), medium as a base pre-trained model for all our experiments. Experimental hyperparameters are presented in App. C.

5.1 Analyzing Temporal Drift Nature of data

We demonstrate the natural drift inherent in the data by analyzing the model's performance over time. To showcase this, we chronologically split each dataset into three sections: training, validation, and test. For the ECHR dataset, the time splits for training, validation, and test are 1960-2016, 2017-2018, and 2019-2022, respectively. For the EU dataset, the time splits are 1953-2012 for training, 2012-2015 for validation, and 2015-2022 for testing. We further divide the training data into two versions: (i) the first half of the chronologically sorted training data, referred to as *Old*, and (ii) the latter half, referred to as *Recent*.

To assess the impact of temporal drift, we train two models separately in online fashion using the *Old* and *Recent* splits of the training set and evaluate them on a fixed test set. To eliminate any confounding effects due to the size of the dataset, both models are trained on the same number of instances. We report the perplexity values in Fig. 1b, where lower values indicate better performance. We observe that the *baseline-Recent* consistently outperforms (i.e., exhibits lower perplexity) the *baseline-Old* across both datasets. This indicates that models trained on data temporally closer to the test set tend to yield superior results, thereby confirming the presence of temporal drift in the data and the need for temporal adaption of models. Additionally, we train a model using the entire training set, referred to as *Baseline-Full*. This model demonstrates enhanced performance across both datasets, which suggests that while recent data provides an advantage, it is also crucial to retain older data for better temporal generalizability. Overall, the performance across all models degrades as we progress through the years in the test split across both datasets, affirming the presence of temporal drift. This observation underscores the necessity of regularly updating models with evolving data to mitigate the effects of temporal degradation on model performance.

To further characterize temporal drift, we measure distributional shifts using Jensen-Shannon divergence, comparing the distribution of vocabulary from the dataset across different time splits. As shown in Table 1a, temporally closer splits exhibit lower divergence scores compared to others, which confirms that these datasets indeed experience temporal drift. However, it is important to note that this analysis might underestimate the full impact of drift, as it focuses solely on lexical-level changes without capturing semantic shifts over time—such as changes in the associated meaning or contextual usage of specific words.

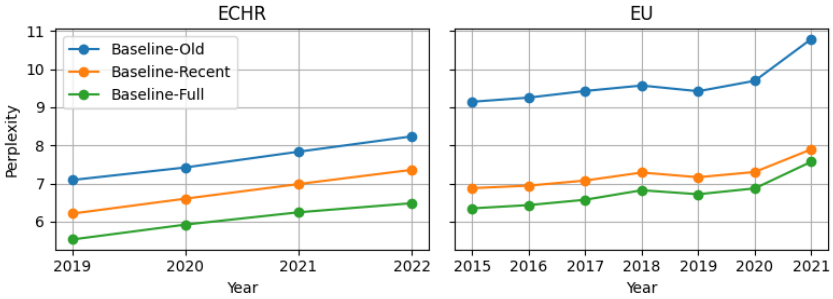
5.2 Baselines & Comparison

FineTune: We initialize GPT-2 medium model and continue fine-tuning whole model in online fashion on incoming data throughout.

We leverage different continual learning algorithms, which are proposed to accumulate knowledge incrementally without forgetting information from previous steps referred to as catastrophic forgetting, to our framework of a boundary-unaware, non-stationary temporal shift setting, treat each new data instance as boundary.

	ECHR	EU
Old	0.23	0.36
Recent	0.15	0.26

(a) Jensen–Shannon divergence score between the split of training set (Old/Recent) and the test set over the vocabulary distribution (x). Higher Score indicates more divergence from the test set distribution.



(b) Impact of temporal drift on the model performance.

Figure 1: Analysis of Temporal Drift Characteristics in the Data.

EWC: (Kirkpatrick et al., 2017) adds a temporal regularization term to the actual loss so that the parameter update from t-1 to t is restricted to avoid over-fitting. It uses Fisher Information Matrices to estimate the importance of parameters to apply a weighted penalty such that the more important parameters to the previous timestamp will have larger penalty weights, balancing the trade-off between previous knowledge and new knowledge. We use online version of EWC (Schwarz et al., 2018).

RecAdam: (Chen et al., 2020) is another regularization technique which modifies Adam optimizer by decoupling the gradient of the quadratic penalty and the annealing coefficient, to preserve the models parameters when updating on new data.

ER: (Rolnick et al., 2019) Experience Replay falls into the category of rehearsal-based methods that stores samples from previous time stamps into a growing memory module. We use a small subset of data randomly sampled from the memory to periodically (every k time steps) retrain the model.

Biased Reservoir Replay: (Lin et al., 2021) While ER uniformly samples from all data encountered in the stream to populate the memory, this strategy biases to store recent samples to in the memory.

MaxLoss Replay: (Lin et al., 2022) Rather than randomly sampling replay examples from memory pool, they sub-sample the ones that have largest losses on the current model, with intent to replay the most forgettable examples, conditioning on the current information (Aljundi et al., 2019).

While all these methods involving fine-tuning the whole model, which turns infeasible with growing model sizes and parameter-efficient strategies have been applied for continual learning, given strong generalization capabilties of base model.

Adapters: (Houlsby et al., 2019) freezes the parameters of the pre-trained model and injects two small modules with up- and down- projection between the self-attention and the feed-forward sub-layers inside each transformer layer sequentially.

LoRA: (Hu et al., 2021) freezes the original parameters of the pre-trained model and introduces trainable low-rank matrices and combines them with the original matrices in the multi-head attention and are updated during fine-tuning.

Prefix Tuning: (Li and Liang, 2021) injects trainable prefix vectors into keys and values of the attention head input and are optimized with reparameterization via a multilayer perceptron.

We evaluate the perplexity score of each document in an online setting before training, and report the averaged scores across the entire corpus in Table 1. A lower perplexity score indicates better performance. All continual learning approaches outperform standard fine-tuning (FT), suggesting an effective balance between stability and plasticity is essential for temporal generalization. Among the regularization-based methods, no single approach emerges as a clear winner, although they both surpass standard fine-tuning, indicating that regularization can mitigate catastrophic forgetting to some extent. Rehearsal-based methods, however, demonstrate superior performance compared to regularization techniques, underscoring the importance of retaining older documents to enhance temporal generalization, as these methods effectively preserve past knowledge. MaxLoss approach, which focuses on replaying the most challenging forgotten examples, further improves retention. BRR method incorporates a recency bias in its sampling strategy, showing that recent documents are particularly valuable for interpolation into future contexts. The success of these rehearsal-based approaches highlights the potential for further research on more effective strategies for selecting the most influential memory pool and optimizing sampling techniques for replay, to enhance temporal generalization.

	ECHR	EU
FT	3.22	5.28
EWC	2.6	5.14
RecAdam	2.74	5.07
ER	2.52	4.77
BRR	2.45	4.63
MaxLoss	2.48	4.71
LoRA	2.63	4.82
PrefixTuning	2.6	4.86
Adapters	2.53	4.72
Ours (DMoE)	2.46	4.65

Table 1: Perplexity scores of different approaches.

	ECHR	EU
Stack	2.59	4.81
Fusion	2.62	4.84
Ensemble	2.52	4.71
Merge	2.5	4.69
Fixed Interval	2.5	4.69
Ada. self-exp.	2.48	4.67
Uniform	2.5	4.7
Exponential	2.48	4.67
DBSCAN	2.46	4.65

Table 2: Ablation Analysis of DMoE: Perplexity Scores

	FT	BRR	Adapters	DMoE
R-1	24.8	32.8	29.7	31.6
R-2	6.6	17.8	12.6	16.2
R-L	16.4	31.5	25.3	29.3
AlignS	41.52	69.53	59.24	65.33
Coher.	42.4	57.5	54.1	56.2
Fluency	59.9	66.5	67.4	68.1

Table 3: Open-ended Generation Analysis on ECHR

While rehearsal-based methods require complete fine-tuning of all parameters, parameter-efficient approaches prove competitive and often outperform regularization methods. Notably, Adapters consistently deliver strong performance across both datasets, although they still lag behind rehearsal-based methods. This suggests that Adapters may suffer from a capacity bottleneck, limiting their ability to accommodate new instances while retaining past knowledge. Our self-expansion strategy mitigates this limitation by dynamically adding new experts as needed and leveraging all experts through similarity-based routing for better aggregation. This is reflected in the superior performance of DMoE approach, competitive with the best rehearsal-based method, BRR, but without the need for storing past data. D-MoE achieves this with minimal parameter updates—just 0.8% of the total trainable parameters compared to the base model or BRR—and avoids training overhead due to BRR's periodic rehearsal, which requires 30% more training data with 100% parameter updates. Despite a slight increase in inference latency (1.12x compared to 1x for BRR), D-MoE strikes an effective balance between efficiency and performance, making it a computationally lightweight yet robust alternative. Moreover, BRR due to its strategy to mix past data for generalization might hurt the correct temporal sequence of legal shifts and can fail to capture evolving jurisprudence, instead reinforcing older precedents that should have been superseded as observed in our detialed case study in App. D.

5.3 Ablation Study

We conduct ablation studies to evaluate design choices regarding when and how to add new experts, as well as to aggregate them for inference.

How to add experts? The modular nature of Adapters allows for the utilization of multiple adapters to capture distinct knowledge through adapter compositions. We explore the following approaches: (i) Stacking, as demonstrated in MAD-X (Pfeiffer et al., 2020c), where multiple adapters are stacked sequentially. This method enables new adapters to learn from prior information as it flows through the stacked adapters and this helps in detecting potential drifts and accumulating new knowledge while relying on older adapters for unchanged information. (ii) Fusion (Pfeiffer et al., 2020a), which involves learning independent adapters in parallel and then combining them through a parametric key-value-based fusion layer. This approach overcomes the limitation of the stacking mechanism, where inference time increases due to the sequential processing of adapters, by expanding experts in parallel instead of in series. (iii) Ensemble (Wang et al., 2021), which aggregates the output predictions of multiple adapters to improve performance, without any learnt fusion layer. (iv) Merging Adapters (Wortsman et al., 2022; Ainsworth et al., 2022; Ilharco et al., 2022), which proposes merging adapters directly at the parameter level to create a single adapter, thereby eliminating the need for multiple inference steps. In all these approaches, we add a new expert periodically after every 2000 timesteps. In both ensemble and merging approaches, we preserve the temporal recency property by using exponentially decaying weights for combining them. From Table 2, we observe that stacking and fusion underperform compared to single adapters, which undermines the intended benefits of adding multiple experts. In stacking, adding a new expert in series can lead to overfitting to recent information, thereby overriding older knowledge. Fusion, on the other hand,

tends to bias the learned layer towards newer data, which can result in the complete forgetting of older information. Although ensembling outputs show some improvement, it still lags behind model merging, which effectively leverages prior experts to form a unified expert.

When to add experts? Deciding when to add a new expert is crucial for balancing model capacity and efficiency. We explore two strategies to determine the optimal timing for expert expansion: (i) Fixed interval where new experts are added after every 2000 timesteps. (ii) Adaptive self-expansion strategy that triggers the addition of new experts based on drift detection, by monitoring changes in the model's perplexity. In both approaches, experts are merged with exponentially decaying weights at inference. Adding experts at fixed intervals may lead to unnecessary expansion, resulting in either underutilized experts if the addition frequency is too high, or forgetting older knowledge if the frequency is too low. In contrast, self-expansion strategy ensures that new experts are added only when necessary, effectively responding to real changes in the data distribution. This approach optimizes model capacity and performance, as reflected in the superior results shown in Table 2.

How to aggregate experts? The aggregation of experts plays a crucial role in enhancing the model's performance by effectively directing it to the most relevant experts. We explore three aggregation strategies: (i) Uniform Aggregation, where all experts are combined with equal weights; (ii) Exponential Decay, which prioritizes recent experts by assigning them greater importance; and (iii) DBSCAN-Based Aggregation, where the relevance of each expert to the current test instance is dynamically determined, rather than relying on static weights. In all these approaches, our proposed self-expansion strategy is employed to determine when to add new experts. From Table 2, we see that the uniform aggregation method dilutes the impact of relevant experts, particularly as data evolves. The exponential decay method improves this by prioritizing recent experts, but it may still be suboptimal due to its fixed decay rate. In contrast, the DBSCAN similarity-based routing method offers context-sensitive aggregation, allowing the model to focus on the most relevant experts, leading to better adaptation and performance.

5.4 Analysis: Open-ended Text Generation

Although perplexity is commonly used to gauge progress in language modeling, we also analyze on open-ended generation using 750 ECHR judgments collected from the HUDOC website, official database of ECHR, after our corpus cut-off date of July 2022 and use models that were trained in an online fashion up until that point to generate next paragraph given the previous paragraphs as context that can fit into the length limitation of the model. While generating entire sections would be ideal, evaluating the quality of an entire section against reference text poses significant challenges. Thus, we opted for paragraph-level generation to facilitate a more manageable and accurate evaluation process. During the training process, each paragraph was appended with an `<|endoftext|>`, allowing to consider the generated content up to this marker as the next paragraph.

We assess the quality of the generated paragraphs using: ROUGE-1,2,L (Lin, 2004) for lexical overlap with the reference paragraph, AlignScore (Zha et al., 2023) for factual consistency based on a unified alignment function between the reference and generated text and UniEval (Zhong et al., 2022), a multi-dimensional metric that evaluates coherence and fluency of the generated paragraph with respect to the context. From Table 3, we observe that all continual learning approaches outperform fine-tuning (FT) in open-ended generation. DMoE consistently outperforms single Adapters across all metrics, underscoring the benefit of multiple experts. However, DMoE still underperforms compared to the BRR approach, indicating the potential for further improvements in open-ended generation, using parameter-efficient approaches.

6 Conclusion

We introduce LexTempus, a dynamic mixture of experts model designed to adapt legal language models to the rapidly evolving legal landscape, characterized by frequent changes in statutes, rulings, and societal shifts. LexTempus employs a self-expansion strategy that dynamically adds experts in response to changes in data distribution, with a similarity-based non-parametric router to aggregate knowledge from multiple experts. Experimental results show that LexTempus surpasses traditional fine-tuning, regularization, and parameter-efficient approaches on ECHR and EU case law datasets in both perplexity and text generation met-

rics, while also competing well with replay-based methods that involve full fine-tuning along with periodically replaying past instances. Future research can explore removing experts when unnecessary and pin-pointedly identify specific layers that that necessitate new experts.

Limitations

The effectiveness of LexTempus is validated on ECHR and EU case law datasets. Future research should assess its generalizability across different legal jurisdictions and languages. Due to computational constraints, we employed GPT-2 as the base model, as full fine-tuning and continual learning experiments require substantial resources. While larger models could offer additional insights, such experiments remain beyond our current capacity. Importantly, our deliberate choice of GPT-2 rather than more recent models that operate under similar computational budgets, was motivated by the need to minimize the confounding effects of pretraining memorization. By using an older, smaller model with an earlier training data cut-off, we were able to better isolate the contributions of our continual learning framework. GPT-2's limited exposure to legal materials during pretraining ensured that observed performance gains were not artifacts of memorized knowledge, but instead reflected the model's ability to adapt through structured, incremental updates. This enabled a cleaner and more controlled evaluation, where improvements could be directly attributed to our method. Although larger and more recent models may offer higher absolute performance, they also carry an increased risk of entangled pretraining knowledge, which complicates attribution of learning effects. Future work could explore scaling LexTempus to such models, while developing methodologies to maintain controlled evaluation settings.

This work acknowledges temporal drift in legal text but does not exhaustively analyze its characteristics—such as gradual vs. abrupt shifts—or their impact on model performance. A more detailed study of these shifts could provide deeper insights into legal language evolution. Attributing concept drift in legal datasets to underlying causes is particularly challenging unless one explicitly examines watershed events with substantial legal expertise. For instance, in ECtHR jurisprudence, identifying near-identical cases with different outcomes is difficult, yet such cases would be of great interest to legal scholarship (Santosh et al., 2024c).

Model editing has recently emerged as a promising approach for updating models by directly modifying their parameters to incorporate new information, offering potential relevance to temporal generalizability (Meng et al., 2022; Mitchell et al., 2021). While prior work has focused on inserting discrete factual statements, these methods struggle with context-dependent adaptations that go beyond simple fact updates (Li et al., 2024). Legal language evolves through nuanced reinterpretations, not explicit knowledge replacements, making model editing insufficient for handling gradual legal shifts driven by societal and jurisprudential changes. Moreover, identifying which aspects of prior knowledge require updating remains a fundamental challenge. In contrast, LexTempus employs continual learning, enabling structured and frequent updates through incremental fine-tuning. Unlike model editing, which assumes a clear distinction between old and new knowledge, LexTempus dynamically integrates evolving patterns via expert expansion.

Another class of approaches, such as retrieval-augmented generation (RAG), enhances models by providing relevant information at inference time without modifying their parameters. While effective for real-time accuracy, these methods face significant storage and retrieval challenges, particularly in legal reasoning, which relies on evolving precedent chains rather than static factual knowledge. Unlike general information-seeking queries, legal questions rarely have clear, unambiguous answers, as case law develops through judicial opinions that build upon one another. This complexity complicates retrieval, especially when no single document definitively answers a query. Additionally, document relevance in law is not solely determined by textual similarity; jurisdictional differences, temporal shifts, and conflicting rules further complicate retrieval (Santosh et al., 2025b, 2024b). Textually similar cases may still be misleading or inapplicable, necessitating context-aware mechanisms (Magesh et al., 2024). Adapting RAG for temporal generalizability in legal applications would require a dedicated study on how to update knowledge bases and ensure contextually relevant retrieval over time.

LexTempus represents a first step toward temporally adaptive legal language models, addressing a critical gap in current approaches. Future research could explore advanced expert aggregation,

selective forgetting mechanisms, and hybrid models that combine continual learning with retrieval-augmented strategies for more robust temporal generalization.

Ethics Statement

The datasets used in this study, specifically the ECHR and EU case law datasets, are publicly available and commonly used for research in legal NLP. Although these datasets are not anonymized and contain real names, we do not anticipate direct harm from our experiments. Nevertheless, the use of such sensitive data necessitates careful consideration of privacy, fairness, and ethical implications.

We emphasize that this research does not advocate for replacing legal professionals with AI systems. Instead, our focus is on augmenting human expertise. AI technologies, while promising, carry risks such as bias, misinformation, or stereotyping from datasets that may reflect historical and societal biases. Additionally, AI-generated content might include factual inaccuracies or misinterpretations, particularly in complex legal contexts. Thus, it is crucial to apply rigorous vetting and maintain human oversight when using AI outputs in legal scenarios. Our findings aim to enhance legal text processing for better efficiency and accuracy, but AI should remain a supportive tool, complementing, not replacing, human judgment.

References

Idan Achituve, Idit Diamant, Arnon Netzer, Gal Chechik, and Ethan Fetaya. 2024. Bayesian uncertainty for gradient aggregation in multi-task learning. *arXiv preprint arXiv:2402.04005*.

Oshin Agarwal and Ani Nenkova. 2022. Temporal effects on pre-trained models for language processing tasks. *Transactions of the Association for Computational Linguistics*, 10:904–921.

Samuel K Ainsworth, Jonathan Hayase, and Siddhartha Srinivasa. 2022. Git re-basin: Merging models modulo permutation symmetries. *arXiv preprint arXiv:2209.04836*.

Hasan Abed Al Kader Hammoud, Ameya Prabhu, Ser-Nam Lim, Philip HS Torr, Adel Bibi, and Bernard Ghanem. 2023. Rapid adaptation in online continual learning: Are we evaluating it right? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18852–18861.

Rahaf Aljundi, Min Lin, Baptiste Goujaud, and Yoshua Bengio. 2019. Gradient based sample selection for online continual learning. *Advances in neural information processing systems*, 32.

Spurthi Amba Hombaiah, Tao Chen, Mingyang Zhang, Michael Bendersky, and Marc Najork. 2021. Dynamic language models for continuously evolving content. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 2514–2524.

Magdalena Biesialska, Katarzyna Biesialska, and Marta R Costa-jussà. 2020. Continual lifelong learning in natural language processing: A survey. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6523–6541.

Zhipeng Cai, Ozan Sener, and Vladlen Koltun. 2021. Online continual learning with natural distribution shifts: An empirical study with visual data. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8281–8290.

Ilias Chalkidis, Manos Fergadiotis, and Ion Androutsopoulos. 2021. Multieurlex-a multi-lingual and multi-label legal document classification dataset for zero-shot cross-lingual transfer. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6974–6996.

Ilias Chalkidis and Anders Søgaard. 2022. Improved multi-label classification under temporal concept drift: Rethinking group-robust algorithms in a label-wise setting. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2441–2454.

Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K Dokania, Philip HS Torr, and Marc'Aurelio Ranzato. 2019. On tiny episodic memories in continual learning. *arXiv preprint arXiv:1902.10486*.

Sanyuan Chen, Yutai Hou, Yiming Cui, Wanxiang Che, Ting Liu, and Xiangzhan Yu. 2020. Recall and learn: Fine-tuning deep pretrained language models with less forgetting. *arXiv preprint arXiv:2004.12651*.

Alexandra Chronopoulou, Matthew E Peters, Alexander Fraser, and Jesse Dodge. 2023. Adaptersoup: Weight averaging to improve generalization of pretrained language models. *arXiv preprint arXiv:2302.07027*.

Nico Daheim, Thomas Möllenhoff, Edoardo Maria Ponti, Iryna Gurevych, and Mohammad Emtiyaz Khan. 2023. Model merging by uncertainty-based gradient matching. *arXiv preprint arXiv:2310.12808*.

Matthew Dahl, Varun Magesh, Mirac Suzgun, and Daniel E Ho. 2024. Large legal fictions: Profiling legal hallucinations in large language models journal of legal analysis (forthcoming).

Cyprien de Masson D'Autume, Sebastian Ruder, Lingpeng Kong, and Dani Yogatama. 2019. Episodic memory in lifelong language learning. *Advances in Neural Information Processing Systems*, 32.

Bhuwan Dhingra, Jeremy R Cole, Julian Martin Eisenschlos, Daniel Gillick, Jacob Eisenstein, and William W Cohen. 2022. Time-aware language models as temporal knowledge bases. *Transactions of the Association for Computational Linguistics*, 10:257–273.

Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96, pages 226–231.

Komal Florio, Valerio Basile, Marco Polignano, Pierpaolo Basile, and Viviana Patti. 2020. Time of your hate: The challenge of time in hate speech detection on social media. *Applied Sciences*, 10(12):4180.

Jens Frankenreiter and Julian Nyarko. 2022. Natural language processing in legal tech. *Legal Tech and the Future of Civil Justice (David Engstrom ed.) Forthcoming*.

Judith Gaspers, Anoop Kumar, Greg Ver Steeg, and Aram Galstyan. 2022. Temporal generalization for spoken language understanding. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Track*, pages 37–44.

Kyle Gorman and Steven Bedrick. 2019. We need to talk about standard splits. In *Proceedings of the conference. Association for Computational Linguistics. Meeting*, volume 2019, page 2786. NIH Public Access.

Lin Lawrence Guo, Stephen R Pfohl, Jason Fries, Alistair EW Johnson, Jose Posada, Catherine Aftandilian, Nigam Shah, and Lillian Sung. 2022. Evaluation of domain generalization and adaptation on improving model robustness to temporal dataset shift in clinical medicine. *Scientific reports*, 12(1):2726.

Lin Lawrence Guo, Ethan Steinberg, Scott Lanyon Fleming, Jose Posada, Joshua Lemmon, Stephen R Pfohl, Nigam Shah, Jason Fries, and Lillian Sung. 2023a. Ehr foundation models improve robustness in the presence of temporal distribution shift. *Scientific Reports*, 13(1):3767.

Yue Guo, Chenxi Hu, and Yi Yang. 2023b. Predict the future from the past? on the temporal data distribution shift in financial sentiment classifications. *arXiv preprint arXiv:2310.12620*.

Suchin Gururangan, Mike Lewis, Ari Holtzman, Noah A Smith, and Luke Zettlemoyer. 2022. Demix layers: Disentangling domains for modular language modeling. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5557–5576.

Elad Hazan et al. 2016. Introduction to online convex optimization. *Foundations and Trends® in Optimization*, 2(3-4):157–325.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pages 2790–2799. PMLR.

Beizhe Hu, Qiang Sheng, Juan Cao, Yongchun Zhu, Danding Wang, Zhengjia Wang, and Zhiwei Jin. 2023. Learn over past, evolve for future: Forecasting temporal trends for fake news detection. *arXiv preprint arXiv:2306.14728*.

Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. 2021. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Xiaolei Huang and Michael Paul. 2018. Examining temporality in document classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 694–699.

Yufan Huang, Yanzhe Zhang, Jiaao Chen, Xuezhi Wang, and Diyi Yang. 2021. Continual learning for text classification with information disentanglement based regularization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2736–2746.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. 2022. Editing models with task arithmetic. *arXiv preprint arXiv:2212.04089*.

Kokil Jaidka, Niyati Chhaya, and Lyle Ungar. 2018. Diachronic degradation of language models: Insights from social media. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 195–200.

Joel Jang, Seonghyeon Ye, Changho Lee, Sohee Yang, Joongbo Shin, Janghoon Han, Gyeonghun Kim, and Minjoon Seo. 2022. Temporalwiki: A lifelong benchmark for training and evaluating ever-evolving language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6237–6250.

Joel Jang, Seonghyeon Ye, Sohee Yang, Joongbo Shin, Janghoon Han, KIM Gyeonghun, Stanley Jungkyu Choi, and Minjoon Seo. 2021. Towards continual knowledge learning of language models. In *International Conference on Learning Representations*.

Mali Jin, Yida Mu, Diana Maynard, and Kalina Bontcheva. 2023. Examining temporal bias in abusive language detection. *arXiv preprint arXiv:2309.14146*.

Xisen Jin, Dejiao Zhang, Henghui Zhu, Wei Xiao, Shang-Wen Li, Xiaokai Wei, Andrew Arnold, and

Xiang Ren. 2022. Lifelong pretraining: Continually adapting language models to emerging corpora. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4764–4780.

Zixuan Ke and Bing Liu. 2022. Continual learning of natural language processing tasks: A survey. *arXiv preprint arXiv:2211.12701*.

Zixuan Ke, Bing Liu, Hao Wang, and Lei Shu. 2021a. Continual learning with knowledge transfer for sentiment classification. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2020, Ghent, Belgium, September 14–18, 2020, Proceedings, Part III*, pages 683–698. Springer.

Zixuan Ke, Hu Xu, and Bing Liu. 2021b. Adapting bert for continual learning of a sequence of aspect sentiment classification tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4746–4755.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526.

Angeliki Lazaridou, Adhi Kuncoro, Elena Gribovskaya, Devang Agrawal, Adam Liska, Tayfun Terzi, Mai Gimenez, Cyprien de Masson d'Autume, Tomas Kocisky, Sebastian Ruder, et al. 2021. Mind the gap: Assessing temporal generalization in neural language models. *Advances in Neural Information Processing Systems*, 34:29348–29363.

Qi Li, Xiang Liu, Zhenheng Tang, Peijie Dong, Zeyu Li, Xinglin Pan, and Xiaowen Chu. 2024. Should we really edit language models? on the evaluation of edited language models. *arXiv preprint arXiv:2410.18785*.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*.

Bill Yuchen Lin, Sida I Wang, Xi Lin, Robin Jia, Lin Xiao, Xiang Ren, and Scott Yih. 2022. On continual model refinement in out-of-distribution data streams. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3128–3139.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Zhiqiu Lin, Jia Shi, Deepak Pathak, and Deva Ramanan. 2021. The clear benchmark: Continual learning on real-world imagery. In *Thirty-fifth conference on neural information processing systems datasets and benchmarks track (round 2)*.

Adam Liska, Tomas Kocisky, Elena Gribovskaya, Tayfun Terzi, Eren Sezener, Devang Agrawal, D'Autume Cyprien De Masson, Tim Scholtes, Manzil Zaheer, Susannah Young, et al. 2022. Streamingqa: A benchmark for adaptation to new knowledge over time in question answering models. In *International Conference on Machine Learning*, pages 13604–13622. PMLR.

David Lopez-Paz and Marc'Aurelio Ranzato. 2017. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Daniel Loureiro, Francesco Barbieri, Leonardo Neves, Luis Espinosa Anke, and Jose Camacho-Collados. 2022. Timelms: Diachronic language models from twitter. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 251–260.

Jan Lukes and Anders Søgaard. 2018. Sentiment analysis under temporal shift. In *Proceedings of the 9th workshop on computational approaches to subjectivity, sentiment and social media analysis*, pages 65–71.

Kelvin Luu, Daniel Khashabi, Suchin Gururangan, Karishma Mandyam, and Noah A Smith. 2022. Time waits for no one! analysis and challenges of temporal misalignment. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5944–5958.

Varun Magesh, Faiz Surani, Matthew Dahl, Mirac Suzgun, Christopher D Manning, and Daniel E Ho. 2024. Hallucination-free? assessing the reliability of leading ai legal research tools. *arXiv preprint arXiv:2405.20362*.

Katerina Margatina, Shuai Wang, Yogarshi Vyas, Neha Anna John, Yassine Benajiba, and Miguel Ballesteros. 2023. Dynamic benchmarking of masked language models on temporal concept drift with multiple views. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2873–2890.

Michael S Matena and Colin A Raffel. 2022. Merging models with fisher-weighted averaging. *Advances in Neural Information Processing Systems*, 35:17703–17716.

Michael McCloskey and Neal J Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier.

Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. 2022. Mass-editing memory in a transformer. *arXiv preprint arXiv:2210.07229*.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D Manning. 2021. Fast model editing at scale. *arXiv preprint arXiv:2110.11309*.

Yida Mu, Kalina Bontcheva, and Nikolaos Aletras. 2023. It's about time: Rethinking evaluation on rumor detection benchmarks using chronological splits. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 724–731.

Anshul Nasery, Jonathan Hayase, Pang Wei Koh, and Sewoong Oh. 2024. Pleas–merging models with permutations and least squares. *arXiv preprint arXiv:2407.02447*.

Cuong V Nguyen, Alessandro Achille, Michael Lam, Tal Hassner, Vijay Mahadevan, and Stefano Soatto. 2019. Toward understanding catastrophic forgetting in continual learning. *arXiv preprint arXiv:1908.01091*.

Yasumasa Onoe, Michael Zhang, Eunsol Choi, and Greg Durrett. 2022. Entity cloze by date: What lms know about unseen entities. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 693–702.

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. 2020a. Adapterfusion: Non-destructive task composition for transfer learning. *arXiv preprint arXiv:2005.00247*.

Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. 2020b. Adapterhub: A framework for adapting transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 46–54.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2020c. Mad-x: An adapter-based framework for multi-task cross-lingual transfer. *arXiv preprint arXiv:2005.00052*.

Ameya Prabhu, Philip HS Torr, and Puneet K Dokania. 2020. Gdumb: A simple approach that questions our progress in continual learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 524–540. Springer.

Yujia Qin, Cheng Qian, Jing Yi, Weize Chen, Yankai Lin, Xu Han, Zhiyuan Liu, Maosong Sun, and Jie Zhou. 2022a. Exploring mode connectivity for pre-trained language models. *arXiv preprint arXiv:2210.14102*.

Yujia Qin, Jiajie Zhang, Yankai Lin, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. 2022b. Elle: Efficient lifelong pre-training for emerging data. *arXiv preprint arXiv:2203.06311*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. 2017. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2001–2010.

Shruti Rijhwani and Daniel Preoţiuc-Pietro. 2020. Temporally-informed analysis of named entity recognition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7605–7617.

David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. 2019. Experience replay for continual learning. *Advances in Neural Information Processing Systems*, 32.

Guy D Rosin, Ido Guy, and Kira Radinsky. 2022. Time masking for temporal language models. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, pages 833–841.

Paul Röttger and Janet Pierrehumbert. 2021. Temporal adaptation of bert and performance on downstream document classification: Insights from social media. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2400–2412.

Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. 2016. Progressive neural networks. *arXiv preprint arXiv:1606.04671*.

TYS Santosh, Mahmoud Aly, Oana Ichim, and Matthias Grabmair. 2025a. Lexgenie: Automated generation of structured reports for european court of human rights case law. *arXiv preprint arXiv:2503.03266*.

TYS Santosh, Kevin D Ashley, Katie Atkinson, and Matthias Grabmair. 2024a. Towards supporting legal argumentation with nlp: Is more data really all you need? *arXiv preprint arXiv:2406.10974*.

TYS Santosh, Rashid Gustav Haddad, and Matthias Grabmair. 2024b. Ecthr-pcr: A dataset for precedent understanding and prior case retrieval in the european court of human rights. *arXiv preprint arXiv:2404.00596*.

TYS Santosh, Tuan-Quang Vuong, and Matthias Grabmair. 2024c. Chronoslex: Time-aware incremental training for temporal generalization of legal classification tasks. *arXiv preprint arXiv:2405.14211*.

TYSS Santosh, Isaac Misael OlguÃn Nolasco, and Matthias Grabmair. 2025b. Lecopcr: Legal concept-guided prior case retrieval for european court of human rights cases. *arXiv preprint arXiv:2501.14114*.

Jeffrey C Schlimmer and Richard H Granger. 1986. Incremental learning from noisy data. *Machine learning*, 1:317–354.

Jonathan Schwarz, Wojciech Czarnecki, Jelena Luketina, Agnieszka Grabska-Barwinska, Yee Whye Teh, Razvan Pascanu, and Raia Hadsell. 2018. Progress & compress: A scalable framework for continual learning. In *International conference on machine learning*, pages 4528–4537. PMLR.

Shai Shalev-Shwartz et al. 2012. Online learning and online convex optimization. *Foundations and Trends® in Machine Learning*, 4(2):107–194.

Chao Shang, Guangtao Wang, Peng Qi, and Jing Huang. 2022. Improving time sensitivity for question answering over temporal knowledge graphs. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8017–8026.

Sidak Pal Singh and Martin Jaggi. 2020. Model fusion via optimal transport. *Advances in Neural Information Processing Systems*, 33:22045–22055.

Anders Søgaard, Sebastian Ebert, Jasmijn Bastings, and Katja Filippova. 2021. We need to talk about random splits. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1823–1832.

George Stoica, Daniel Bolya, Jakob Bjorner, Pratik Ramesh, Taylor Hearn, and Judy Hoffman. 2023. Zipit! merging models from different tasks without training. *arXiv preprint arXiv:2305.03053*.

Fan-Keng Sun, Cheng-Hao Ho, and Hung-Yi Lee. 2019. Lamol: Language modeling for lifelong language learning. In *International Conference on Learning Representations*.

Derek Tam, Mohit Bansal, and Colin Raffel. 2024. Merging by matching models in task parameter subspaces. *Transactions on Machine Learning Research*.

Aman Tiwari, Prathamesh Kalamkar, Atreyo Banerjee, Saurabh Karn, Varun Hemachandran, and Smita Gupta. 2024. Aalap: Ai assistant for legal & paralegal functions in india. *arXiv preprint arXiv:2402.01758*.

Santosh Tyss, Hassan Sarwat, Ahmed Mohamed Abdelaal Abdou, and Matthias Grabmair. 2024. Mind your neighbours: Leveraging analogous instances for rhetorical role labeling for legal documents. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 11296–11306.

Hong Wang, Wenhan Xiong, Mo Yu, Xiaoxiao Guo, Shiyu Chang, and William Yang Wang. 2019. Sentence embedding alignment for lifelong relation extraction. In *Proceedings of NAACL-HLT*, pages 796–806.

Xinyi Wang, Yulia Tsvetkov, Sebastian Ruder, and Graham Neubig. 2021. Efficient test time adapter ensembling for low-resource language varieties. *arXiv preprint arXiv:2109.04877*.

Yaqing Wang, Subhabrata Mukherjee, Xiaodong Liu, Jing Gao, Ahmed Hassan Awadallah, and Jianfeng Gao. 2022. Adamix: Mixture-of-adapter for parameter-efficient tuning of large language models. *arXiv preprint arXiv:2205.12410*, 1(2):4.

Gerhard Widmer and Miroslav Kubat. 1993. Effective learning in dynamic environments by explicit context tracking. In *Machine Learning: ECML-93: European Conference on Machine Learning Vienna, Austria, April 5–7, 1993 Proceedings 6*, pages 227–243. Springer.

Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. 2022. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International conference on machine learning*, pages 23965–23998. PMLR.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin A Raffel, and Mohit Bansal. 2024. Ties-merging: Resolving interference when merging models. *Advances in Neural Information Processing Systems*, 36.

Enneng Yang, Zhenyi Wang, Li Shen, Shiwei Liu, Guibing Guo, Xingwei Wang, and Dacheng Tao. 2023. Adamerging: Adaptive model merging for multi-task learning. *arXiv preprint arXiv:2310.02575*.

Huaxiu Yao, Caroline Choi, Bochuan Cao, Yoonho Lee, Pang Wei W Koh, and Chelsea Finn. 2022. Wild-time: A benchmark of in-the-wild distribution shift over time. *Advances in Neural Information Processing Systems*, 35:10309–10324.

Jaehong Yoon, Eunho Yang, Jeongtae Lee, and Sung Ju Hwang. 2017. Lifelong learning with dynamically expandable networks. *arXiv preprint arXiv:1708.01547*.

Jaehong Yoon, Eunho Yang, Jeongtae Lee, and Sung Ju Hwang. 2018. Lifelong learning with dynamically expandable networks. In *International Conference on Learning Representations*.

Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. 2024. Language models are super mario: Absorbing abilities from homologous models as a free lunch. In *Forty-first International Conference on Machine Learning*.

Yuheng Zha, Yichi Yang, Ruichen Li, and Zhiting Hu. 2023. Alignscore: Evaluating factual consistency with a unified alignment function. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11328–11348.

Jinghan Zhang, Junteng Liu, Junxian He, et al. 2023a. Composing parameter-efficient modules with arithmetic operation. *Advances in Neural Information Processing Systems*, 36:12589–12610.

Yuji Zhang, Jing Li, and Wenjie Li. 2023b. Vibe: Topic-driven temporal adaptation for twitter classification. *arXiv preprint arXiv:2310.10191*.

Zhixue Zhao, George Chrysostomou, Kalina Bontcheva, and Nikolaos Aletras. 2022. On the impact of temporal concept drift on model explanations. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 4039–4054.

Ming Zhong, Yang Liu, Da Yin, Yuning Mao, Yizhu Jiao, Pengfei Liu, Chenguang Zhu, Heng Ji, and Jiawei Han. 2022. Towards a unified multi-dimensional evaluator for text generation. *arXiv preprint arXiv:2210.07197*.

Lee B Ziffer. 2023. The robots are coming: Ai large language models and the legal profession. In *American Bar Association*.

A Related Work

Continual Learning Most research in continual learning (or lifelong learning) initially centered around computer vision tasks, and more recently, it has gained attention in the NLP field (Ke et al., 2021b,a; Biesialska et al., 2020; Ke and Liu, 2022; Sun et al., 2019; Wang et al., 2019). The majority of these works adopted traditional task-incremental, domain-incremental, or label-incremental settings. While our temporal adaptation setting bears resemblance to the domain-incremental setting, a crucial distinction lies in the assumption of strict boundaries in domain incremental settings, which is not applicable to our temporal adaptation where the boundaries between drifts are blurred (Prabhu et al., 2020; Aljundi et al., 2019) and the assumption that incoming samples are from disjoint data distributions is no longer valid (Al Kader Hammoud et al., 2023). Generally, continual learning algorithms can be categorized into (i) Rehearsal-based methods (Rolnick et al., 2019; Rebuffi et al., 2017), which maintain a memory buffer of older data to perform experience replay with actual data (de Masson D'Autume et al., 2019), automatically generated data (Sun et al., 2019), or previously computed gradients (Lopez-Paz and Ranzato, 2017), (ii) Regularization-based approaches (Kirkpatrick et al., 2017; Chen et al., 2020; Huang et al., 2021), which regularizes neural network parameters from drastic updates for new information to preserve the information of older ones, preventing overfitting to the newer data (iii) Network expansion methods (Qin et al., 2022b; Gururangan et al., 2022; Yoon et al., 2018) which dynamically grow branches to accommodate for newer incoming data.

Traditionally, CL has been evaluated in offline settings where models can revisit all samples within the current task. However, recent advancements in computer vision have shifted focus towards online CL, aiming for rapid model adaptation with new incoming single instance (Al Kader Hammoud et al., 2023; Prabhu et al., 2020). Recent works like CLOC (Cai et al., 2021) and CLEAR (Lin et al., 2021) have explored CL in a streaming fashion, where data arrives in a sequence ordered by timestamps, creating a temporal stream of evolving visual concepts with gradual, boundary-unaware distribution shifts. They focus on the model's ability to adapt quickly and extrapolate into the future by evaluating its performance on future data. Our work extends this temporal streaming setup in an online fashion to legal language models, enabling them to continuously adapt to new data distributions and improve generalization on future data.

Model merging combines multiple pretrained models into one model through a weighted averaging of their parameters, giving it the combined abilities of each individual model without any additional training. Unlike model ensembling, which combines model outputs and typically increases inference costs, model merging directly combines parameters, maintaining performance across all merged models without adding inference overhead. Several weighting schemes have been explored such as task arithmetic with simple averaging (Ilharco et al., 2022; Wortsman et al., 2022), Fisher-weighted merging (Matena and Raffel, 2022), RegMean (Tam et al., 2024), Git Re-Basin (Ainsworth et al., 2022), TIES-Merging (Yadav et al., 2024), Ada-merging(Yang et al., 2023), uncertainity based merging (Daheim et al., 2023; Achituve et al., 2024), PLeaS-Merging (Nasery et al., 2024), Zipit (Stoica et al., 2023), OT-Fusion (Singh and Jaggi, 2020), DARE (Yu et al., 2024). While initially developed for whole model merging, these techniques have been extended to parameter-efficient modules (Zhang et al., 2023a; Chronopoulou et al., 2023; Qin et al., 2022a), which differs from earlier approaches like AdapterFusion (Pfeiffer et al., 2020a) and mixture-of-experts (Wang et al., 2022) that combine module outputs with additional training. These prior methods have been studied in multi-task setups to create one unified model with improved cross-task generalization. In contrast, our approach dynamically generates a merged model for each test instance on the fly, using a non-parameterized similarity-based router.

B Dataset

Dataset	ECHR	EU
Train (Old)	5,808	11,352
Train (New)	5,807	11,352
Validation	1,688	2,237
Test	2,426	4,915
Total	15,729	29,856

Table 4: Dataset statistics for ECHR and EU case law.

C Implementation Details

Baseline FT models We train the models using negative log-likelihood loss using AdamW optimizer (Loshchilov and Hutter, 2017) and with a learning rate of 5e-5, momentum of 0.9, and weight decay of 3e-7. We employ a sliding window of 1024 tokens with a stride of 512 where the first 512 tokens as context. We use a batch size of 4 for both datasets. We use 16-bit automatic mixed precision and gradient accumulation to accelerate training and save memory. All the experiments were performed on a GPU cluster with NVIDIA A40 48GB PCIe 4.0.
EWC (Kirkpatrick et al., 2017) We use the online version of EWC training to avoid memory overflow in the computation of Fischer information matrices, as detailed in (Kirkpatrick et al., 2017). We set λ of 0.5 which controls the strength of regularization-based EWC loss with a decay term for older data γ set to default of 1.0.
RecAdam (Chen et al., 2020) We use Adam's epsilon of 1e-6 and sigmoid as the annealing function
ER (Rolnick et al., 2019) We replay a set of 3 documents by random sampling from an evolving memory module after every 10 training documents.
Biased Reservoir Replay (Lin et al., 2021) Our reservoir is populated based on recency bias with 10%, 20%, 30%, 40% from the time-based quartiles of the previous data.
MaxLoss Replay (Lin et al., 2021) It replays 3 examples with highest losses from 10 random sampled examples.

We use AdapterHub framework (Pfeiffer et al., 2020b) to implement Adapters, LoRA, and Prefix Tuning.
Adapters (Houlsby et al., 2019) We use bottleneck adapters with a reduction factor of 64.
LoRA (Hu et al., 2021) We set the rank r to 16.
Prefix Tuning (Li and Liang, 2021) We use a prefix length of 30.

DMoE: The running statistics of mean and standard deviation of test loss are maintained with sliding windows of length 100 to compute z-score statistic. We set threshold for drift trigger to be -1.4 in ECHR and -1.15 in EU. We use 100 as neighboring data points to form a cluster and maximum distance value to be 10 in EU and 4 in ECHR.

D Case Study

Case Study A: Temporal Adaptation in Gender Identity Jurisprudence To demonstrate the effectiveness of our model, we analyze L. v. Lithuania (no. 27527/03), an ECHR case concerning the legal recognition of transgender individuals' gender identity. This case arose amid evolving societal and medical perspectives on gender, reflecting a broader shift in human rights jurisprudence. Earlier rulings, such as Rees v. UK (1986), Cossey v. UK (1990), and Sheffield and Horsham v. UK (1998), upheld a biological definition of gender, deferring to States' margin of appreciation in regulating marriage and identity laws. However, in Christine Goodwin v. UK (2002), the Court overturned this stance, ruling that denying legal gender recognition violated Articles 8 and 12 in light of contemporary medical and societal developments. This decision marked a turning point, emphasizing individual rights over state discretion in gender identity recognition.

When generating reasoning for L. v. Lithuania, LexTempus correctly prioritized the updated precedent from Christine Goodwin (2002): "In light of Christine Goodwin v. UK (2002), the refusal to provide a clear legal framework for gender reassignment violates Articles 8 and 12, as it no longer reflects contemporary societal and medical standards." By contrast, the Replay-Based Model (BRR) failed to account for this legal shift, producing: "The refusal to establish a clear legal framework for gender reassignment falls within the respondent State's margin of appreciation, as upheld in Cossey v. UK (1990)." This discrepancy arises because ER-based models (including Biased Reservoir Replay) mix past and present case law without maintaining the correct temporal sequence of legal shifts. By over-relying on uniform or biased sampling strategies, replay-based methods risk reinforcing outdated precedents, failing to capture the progressive nature of jurisprudence. LexTempus, by contrast, maintains jurisprudential consistency, ensuring alignment with modern human

rights interpretations. This case study underscores the necessity of adaptive, non-replay-based continual learning strategies in legal AI, ensuring models accurately reflect temporal legal shifts rather than perpetuating outdated legal standards.

Case Study B: Evolving Hate Speech Jurisprudence Under Article 10 The European Court of Human Rights (ECHR) has progressively refined its stance on freedom of expression (Article 10) and the prohibition of hate speech, adapting to shifting societal norms. Early cases, such as Jersild v. Denmark (1994), prioritized journalistic freedom, ruling that a journalist who broadcast racist remarks was not liable, as his intent was to report rather than endorse the statements. Similarly, in Perinçek v. Switzerland (2015), the Court ruled that a Turkish politician's denial of the Armenian genocide did not constitute hate speech under Article 10, as it did not incite violence or hatred. However, a stricter standard emerged in Lilliendahl v. Iceland (2020), where a man was convicted for homophobic remarks, with the Court holding that statements "promoting intolerance" could be lawfully restricted. This shift reflects the Court's increasing recognition of the harm caused by discriminatory speech, particularly in the context of marginalized groups.

To evaluate temporal generalizability, we tested LexTempus and a Replay-Based Model (BRR) on a hypothetical case where a politician made inflammatory anti-LGBTQ+ statements in a public speech. LexTempus correctly prioritized the most recent precedent, aligning with Lilliendahl v. Iceland (2020) by reasoning: "Under Lilliendahl v. Iceland (2020), public statements promoting intolerance can justify legal restrictions under Article 10." By contrast, the BRR model incorrectly referenced older jurisprudence, generating: "Following Perinçek v. Switzerland (2015), political speech enjoys strong protections, even when controversial." This discrepancy highlights a fundamental challenge in legal NLP—models that rely on replay-based learning risk blending outdated and current case law inconsistently, leading to erroneous legal conclusions. LexTempus, by dynamically adapting to evolving legal interpretations, ensures its reasoning aligns with the most relevant legal standards, demonstrating the critical need for adaptive learning in legal language models.