**UDC: 004.853:519.722** **doi: 10.32620/reks.2026.2.07**

**Arsentii KRIUKOV, Yurii PUSHKARENKO**

***Taras Shevchenko National University of Kyiv, Kyiv, Ukraine***

# ADAPTIVE KNOWLEDGE REGULARIZATION FOR CONTINUAL LEARNING IN TRANSFORMER ARCHITECTURES

***The subject*** *matter of the article is development of latent representation regularization mechanism in transformer-based architecture under conditions of continuous learning with domain shifts. Modern language models achieve high quality in static learning scenarios, but they remain limited in long-term operation cases, incremental adaptation to new domains and lacking resistance to catastrophic forgetting, especially in the absence of access to previously observed data. This paper explores the possibility of overcoming these limitations by combining uncertainty inspired regularization and forgetting attention mechanisms into one transformer architecture.* ***The goal of the study*** *is to design, implement and validate transformer-based architecture with multi-level representation regularization mechanism that can help transformer-based language models efficiently adapt to alternative data distribution while retaining previously acquired knowledge. The proposed approach aims to achieve an equilibrium between model adaptability and stability in continual learning without requiring complete model retraining or legacy data retention.* ***The tasks to be solved*** *in this study include: formalization of an unified conceptual latent representations regularization method that combines Bayesian uncertainty inspired latent representations regularization with head-wise attention scaling in attention mechanism; implementation of this method into the transformer model; creating the experimental case of continuous language modeling with sequential domain shifts; give quantitative estimation of model forgetting and stability in prediction quality; compare the proposed model to classical naive fine-tuning, LoRA and parameter regularization methods.* ***The conclusions*** *demonstrate that the proposed method achieves lower forgetting, lower perplexity on previously learned domains and a better stability–plasticity trade-off than naive fine-tuning, LoRA and Elastic Weight Consolidation, while requiring comparable computational resources.* ***The scientific novelty*** *of proposed approach consists in development of layer-selective latent regularization framework for continual language modeling which integrates an attention with forgetting mechanism with preserving domain-invariant representations through statistical alignment in latent space for reducing forgetting in continual learning scenarios. Unlike existing approaches to continual learning that consider model regularization either on parameter or memory level (by using previous data), the proposed approach moves regularization into representation space, where it uses both direct regularization via proposed uncertainty-based regularization and indirect regularization via attention with forget gate, ensuring the models' possibility of stable continual language modeling in non-stationary environments.*

**Keywords***: continual learning; catastrophic forgetting; latent space; attention mechanism; uncertainty; language models; natural language processing, transformers, information entropy.*

## 1. Introduction

### 1.1 Motivation

The research presented in this paper belongs to the field of computer science, in particular to the area of machine learning, natural language processing, continuous learning. From applied point of view, the work results are relevant for artificial intelligence systems (AI agents) that are working in continuous operating, such as dialogue agents, decision support systems, automated analytical platforms. These systems are expected to function over long periods of time, interacting with evolving data and adapting to changing tasks and domains.

Relevance of the problem is caused by growing need of AI systems, especially Large Language Model-based, that are suitable for long-term autonomous operation, incrementally learning from streaming data and stable knowledge retention throughout entire operation cycle. The absence of controllable regularization mechanisms is one of the main causes of model performance degradation over time. As a result, models require retraining on large datasets, that costs money, is time-consuming and intelligent services, based on such kind of model, are less reliable in long-term deployment scenarios.

### 1.2 State of the art

The problem of catastrophic forgetting in neural networks is one of the major problems of machine learning systems that are working in dynamic environments and need to constantly adapt to new data and tasks [1]. First, specification of the problem. Phenomenon of catastrophic forgetting happens when models' performance

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

on their previous knowledge drastically decreases when model adapts to new knowledge.

Modern studies, oriented on reducing catastrophic forgetting in continual learning are commonly categorized into four major groups: replay-based methods, parameter regularization methods, parameter isolation methods and hybrid approaches that combine elements from other methods, listed above. [2]

Replay methods based on storing legacy learning data or reproduction of this data using generative models (generative replay). [3] However they suffer from scalability issues, privacy concerns and storage overhead. And for generative replay, training process of such models is complicated and requires a lot of resources. Also, in transformer-based continual learning scenarios, use of replay may require storing large token buffers that is impractical. One of the recent improvements of this type of methods is Selective Attention-Guided Knowledge Retention, approach [4], that uses selective attention distillation to improve efficiency of data usage allowing of usage only fraction of replay samples. While replay methods remain among the most effective empirically, their reliance on stored or generated data limits scalability in natural language processing in continual learning scenarios.

Regularization methods, on the other hand, are controlling not model memory, but its parameters and how they are updated. These methods constrain parameter updates to preserve knowledge from previous tasks. In such methods there is no need for data storage and the principle used to reduce forgetting is protecting task-specific weights. For example, Elastic Weight Consolidation (EWC) [5] uses Fisher Information as parameter importance estimate and during training model is punished if this parameters changed a lot. But such approach often trades plasticity for stability and is potentially limited to certain number of tasks (when all model parameters are constrained already, model would not learn efficiently on new tasks). Also, in high -dimensional transformer models parameter importance estimates may be unreliable.

Parameter isolation methods allocate separate parameters for each domain or task to avoid overwriting task-specific parameters. In context of transformer models, this approach takes form of adapters [6], LoRA modules [7] or prompt-based components added per task. This approach provides strong task separation and is compatible with large pretrained transformer models via parameter-efficient tuning. One of the recent papers using this approach is Adapter-Based Multi-Modal Continual Learning with a Two-Stage Learning Strategy [8]. Overall, adapters and related parameter isolation strategies are particularly suitable for natural language processing in continual learning scenarios, but adding new tasks for model may require architectural expansions and finding balance between knowledge transfer across tasks and isolation remains an active challenge.

Hybrid approaches combine elements from replay, regularization and isolation to leverage complementary strength. These models benefit in memory usage balancing, stability control and task adaptation. For instance, method, Averaged Gradient Episodic Memory [9] combines replay with gradients regularization to prevent degradation of performance on previously learned tasks. Similarly, Dark Experience Replay [10] uses replay buffers alongside regularization of logits to stabilize models prediction capabilities across different tasks. These methods often achieve state-of-art empirical performance that address different aspects of catastrophic forgetting and represent one of the most promising directions in solving this problem.

The proposed model introduces a representation-level regularization mechanism that differs fundamentally from existing replay-based, parameter regularization-based and parameter isolation-based approaches. Unlike parameter regularization approaches which constrain individual parameters based on estimated importance, the proposed method stabilizes latent feature distributions in transformer layers. This prevents semantic drift, avoiding unreliable parameter importance estimation for high-dimensional transformer models, allowing the model to maintain domain-invariant semantic structures while still permitting parameter adaptation. Usage of forget gate in attention mechanism can be represented as a method of controlled sharing of knowledge between domains.

### 1.3 Objectives and tasks

Formulation of the problem: develop and experimentally validate a continual learning method for transformer-based language models that reduces forgetting while preserving adaptation to new domains.
Research objectives:

1. Analyze current approaches to the problem of catastrophic forgetting in transformer language models and to methods of continual learning.
2. Develop uncertainty-inspired regularization mechanism suitable for transformer architecture.
3. Integrate a contextual forgetting mechanism based on Forgetting Attention with proposed regularization method.
4. Implement a transformer-based language model with the proposed multi-layer regularization mechanism using PyTorch framework.
5. Experimentally evaluate the proposed method by measuring perplexity, forgetting, stability–plasticity trade-off, and latent representation drift under sequential domain adaptation conditions.

6. Compare the proposed method with naive fine-tuning, EWC and LoRA in terms of perplexity, forgetting, computational cost, and stability–plasticity trade-off.

The article is structured as follows:

Section 2 describes the materials and methods used in this study, including: formal definition of the problem, the description of the proposed representation-level regularization framework, the FoX-inspired attention mechanism, the model architecture and hyperparameters, used in experiments, data description and preparation, experimental setup, approaches, that are chosen for comparison with proposed method. Section 3 shows results of the experimental evaluation and case-study, including perplexity dynamics, forgetting metrics, stability-plasticity analysis, layer-wise representation drift. Here is also provided a detailed discussion of the obtained results, their meaning, and their comparison to existing approaches.

Section 4 concludes the article by summarizing the main findings and outlining directions for future research.

## 2. Materials and methods of research

First, formalization of the task. Model is working in continuous learning scenario, sequentially learning on sequence of text data D, $D = \{D_1, D_2, \dots, D_N\}, D_i \cap D_j = \emptyset, \forall i, j = \overline{1,N}, i \neq j$. At the learning step i model have access only to the dataset $D_i$, it does not have access to previously observed datasets $\{D_1, D_2, \dots, D_{i-1}\}$. In terms of evaluation models` efficiency, the goal is to minimize models` perplexity on current dataset $D_i$ and at the same time minimize perplexity degradation on previous datasets $\{D_1, D_2, \dots, D_{i-1}\}$. From optimization perspective, the goal is to obtain during model training a sequence of parameter states $\theta_1, \theta_2, \dots, \theta_N$, where $\theta_i$ stands for models` parameter set after model finished training on dataset $D_i$. At each learning step, the task-specific loss function is defined (1).

$$L_i(\theta_i) = E_{(x,y)\sim D_i}\left[l\left(f_{\theta_i}(x), y\right)\right], \text{for } i = \overline{1,N}, \quad (1)$$

where $D_i$ is data distribution on current learning step

E is the expectation operator over samples from $D_i$;

y is the target sequence corresponding to the input x;

l is per-sample loss function;

$f_{\theta_i}$ is language model with parameters $\theta_i$.

The dataset that is available to model during training step is a finite sample drawn from unknown distribution. $L_i(\theta_i)$ represents expected task-specific objective at training step i. Minimizing $L_i(\theta_i)$ is equivalent to minimizing model perplexity on current dataset. But, when optimizing only $L_i(\theta_i)$ on each training step, this works only for current dataset and can and definitely will lead to changing parameters that are vital for previous tasks, and results in catastrophic forgetting. This is the general issue in continual learning, especially severe in transformer-based language models. [5],[11] Assume, that $R_i$ as metric of previous tasks knowledge retention. Then, to optimize the model during all learning stages, there is a need to minimize losses on all previous tasks, so, $R_i(\theta_i)$ (2) will be calculated as following.

$$R_i(\theta_i) = \frac{1}{i-1}\sum_{k=1}^{i-1} L_k(\theta_i), \quad (2)$$

where $L_k(\theta_i)$ is task-specific loss function.

But on step i model do not have access to $\{D_1, D_2, \dots, D_{i-1}\}$, so it can not directly minimize $R_i(\theta_i)$. To solve this problem Bayesian approach to continual learning was used as an inspiration. In Bayesian approach, model parameters are not deterministic values, they are described by posterior distribution $p(\theta|D)$. This constrains parameter updates by incorporating the posterior from previous tasks as a prior for the current task. In classic approach, model parameters are deterministic values. In transformer models knowledge is stored as contextual representations (hidden state), formed by self-attention mechanism. [12] When the parameters are updated, even a little, representations can change a lot. Therefore, constraining representation drift can be used as indirect way of stabilizing model parameters. Assuming this, the proposed approach is formulated. It is conceptually related to Bayesian metaplasticity methods such as Metaplasticity from synaptic uncertainty (MESU) [13], which use synaptic uncertainty to regularize parameter updates and minimize forgetting. In MESU uncertainty is associated with network parameters and used as estimate of parameter importance. In this approach plasticity is modulated via adaptive learning rate that is proportional to parameter variance. However, for transformer-based language models, using probabilistic parameters is computationally infeasible due to large number of parameters, high complexity of estimating the full posteriors and unstable optimization in high-dimensional weight spaces. On the other hand, approach proposed in this paper uses uncertainty idea not on model parameter space, but on latent representation space. Instead of controlling distribution over weights, proposed approach is a representation-level analog of MESU idea and here variance of latent activations is an indicator of knowledge stability.

### 2.1 Representation level regularization

Assume that transformer model has L layers. In this case for input sample x, initial representation $h^{(0)}$ is defined as (3):

$$h^{(0)} = x. \quad (3)$$

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

The output $h^{(l)}$ of the transformer block l then is (4):

$$h^{(l)} = f_\theta^{(l)}\left(h^{(l-1)}\right), \quad (4)$$

where $f_\theta^{(l)}$ is transformation made by transformer block l. For realization of proposed approach, representations are extracted after the second LayerNorm operation (5) inside each transformer block in proposed architecture:

$$z^{(l)} = LN2\left(\tilde{h}^{(l)}\right), \quad (5)$$

$$\tilde{h}^{(l)} = h^{(l-1)} + MHSA\left(LN1\left(h^{(l-1)}\right)\right), \quad (6)$$

where $z^{(l)}$ is the latent representations being regularized;

$\tilde{h}^{(l)}$ is the output of the first residual add operation;

MHSA returns the output of Multi-Head Self Attention mechanism;

LN1 and LN2 are layer normalization operations;

FFN returns output from Feed Forward Network module.

The main idea here is to model latent representations $z^{(l)}$ with Gaussian distributions. The rationale is as follows. This is based on Maximum Entropy principle, formulated in Elements of Information Theory [14]. Assume that latent vector $z \in \mathbb{R}^d$ is extracted from one of the transformer layers.

Also assume that the only known information about this vector is its mean (7),

$$E[z] = \mu, \quad (7)$$

and covariance (8).

$$E[(z-\mu)(z-\mu)^T] = \Sigma\,. \quad (8)$$

The Maximum Entropy principle states that among all probability distributions that satisfy known statistical constraints (9) (mean and variance in this case) the one that maximizes entropy should be chosen. So, approximation of latent activation vector z requires to find function (distribution) $q^*(z)$, (10) that maximizes entropy (11):

$$Q = \{q(z) | E[z] = \mu, E[(z-\mu)(z-\mu)^T] = \Sigma\}, \quad (9)$$

$$q^*(z) = \arg\max_{q \in Q} H(q(z))\,, \quad (10)$$

$$H(q) = -\int q(z) * \log q(z) dz\,, \quad (11)$$

where Q is the set of all such distributions $q(z)$ that have the same $\mu$ (defined in (7)) and $\Sigma$ (defined in (8));

$H(q)$ is the entropy of distribution q.

According to the "Elements of Information Theory" this optimal distribution $q^*(z)$ is calculated as in (12):

$$q^*(z) = N(\mu, \Sigma), \quad (12)$$

and it is multidimensional Gaussian distribution.

Full proof of this statement can be found in Elements of Information Theory [14]. In case of latent representations, the distribution with maximum entropy guarantees that no additional unwanted structural assumptions are introduced by this approximation.

So, for each learning step i, representation is a Gaussian distribution $q_i^{(l)}(z)$ (13) with diagonal covariation matrix $\Sigma_i^{(l)}$ (14):

$$q_i^{(l)}(z) = N\left(\mu_i^{(l)}, \Sigma_i^{(l)}\right), \quad (13)$$

$$\Sigma_i^{(l)} = \begin{pmatrix} \sigma_1^2 & \cdots & 0 \\ \vdots & \sigma_2^2 & \vdots \\ 0 & \cdots & \vdots \end{pmatrix}, \quad (14)$$

where $\mu_i^{(l)}$ is mean average of latent representations in layer l after learning on $D_i$;

$\sigma_j^2$ is dispersion (uncertainty) for each dimension j.

The mean and variance of the latent representations at each layer are retained. In (14) diagonal approximation of covariance estimation is used because in high-dimensional representation space full covariance estimation requires $O(d^2)$ parameters, is unstable and computationally infeasible.

To teach the model to retain its previously acquired knowledge, loss function needs to be upgraded: now the model is penalized not only for bad adaptation to new domain via standard loss function, but also for bad retention of previous knowledge. The overall training objective at step i is to minimize the following loss function $L_i$ (15):

$$L_i = E_{(x,y)\sim D_i}[l(f_\theta(x), y)] + \lambda \sum_{l=1}^{L} KL(q_i^{(l)}(z) \,\|\, q_{i-1}^{(l)}(z)], \quad (15)$$

where KL is Kullback-Leibler distance.

In this case, KL from (15) can be calculated as follows (16) (according to (14)):

$$KL(q_i^{(l)}(z) || q_{i-1}^{(l)}(z)) = \frac{1}{2}\sum_{j=1}^{d}[\log\frac{\sigma_{i-1,j}^2}{\sigma_{i,j}^2} + \frac{\sigma_{i,j}^2 + (\mu_{i,j} - \mu_{i-1,j})^2}{\sigma_{i-1,j}^2} - 1]. \quad (16)$$

The $\lambda$ parameter here controls the strength of the regularization penalty for information loss. The idea of using KL divergence here is inspired by Variational Continual Learning [11], where it is used as a measure of posterior divergence across tasks.

In this paper, similar ideas are applied, but to the latent space. Assume that on training step $i-1$ on layer l latent representations are represented as following (17)

$$q_{i-1}^{(l)}(z) = N\left(\mu_{i-1}^{(l)}, \Sigma_{i-1}^{(l)}\right). \quad (17)$$

To minimize forgetting, the distribution on step i have to be very close to the distribution on step $i-1$ (18),

$$q_{i-1}^{(l)}(z) \approx q_i^{(l)}(z), \quad (18)$$

which is equivalent to minimizing (19):

$$KL(q_i^{(l)}(z)||q_{i-1}^{(l)}(z)). \quad (19)$$

Here KL divergence is a measure of distance between distributions $q_{i-1}^{(l)}(z)$ and $q_i^{(l)}(z)$. Minimizing KL divergence also can be interpreted as informational projection of the new distribution $q_i^{(l)}(z)$ on information space of the old distribution $q_{i-1}^{(l)}(z)$.

The KL divergence penalty in latent space (19) can be also interpreted as analogue to Fisher information penalty (which is used by EWC in parameter space). It provides computationally effective approximation to Bayesian uncertainty. What is done above is adaptation of Bayesian uncertainty regularization approach, proposed in MESU [13] and Variational Autoencoders [15] to the transformer-based language models by introducing a layer-wise regularization term over latent representations.

## 2.2 FoX-inspired attention

While MESU-inspired distributional regularization approach preserves knowledge across tasks, it does not control how the model attends to its past content. In transformer language models self-attention is a core driver of context modeling, it determines how information from previous tokens affects current predictions. Even with stabilized representations standard self-attention may overweight certain past tokens, and this can potentially interfere with learning new domain. To overcome this limitation, the proposed approach incorporates the Forgetting Attention (FoX) proposed in [16] into the attention computation. This mechanism augments standard softmax attention (20) with dynamic forget gate, allowing the model to control influence of past tokens in context-dependent, learnable manner. This gate is computed using current hidden states and affects softmax attention scores before applying weighted sum. Below is a brief description of this mechanism. Head-wise multiplicative forgetting modulation is applied directly to the attention weights (21):

$$A(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right), \quad (20)$$

$$A_{FoX}(Q, K, V) = A(Q, K, V) * g(x), \quad (21)$$

where $A(Q, K, V)$ is standard attention head output;

$A_{FoX}(Q, K, V)$ is FoX mechanism attention head output;

$g(x) = \sigma(W_g * x + b)$ is learnable gate function that produces head-specific modulating coefficients.

By modulating attention pathways FoX reduces gradient interference across tasks which complement MESU-inspired distributional regularization.

## 2.3 System architecture

### 2.3.1 Transformer block

Let's look at architecture of transformer block l of proposed model (Fig. 1). It is based on transformer block architecture, proposed in [12]. It has Pre-LN configuration, where normalization is applied before the attention mechanism, to improve training stability and gradient flow [17]. Input of this block is hidden representation $h^{(l)}$. Then this representation is normalized by using a layer normalization operator (LayerNorm1), which stabilizes it.

The normalized representation is given to the modified multi-head self-attention module with forget gate. In this module, standard scaled dot-product attention augmented with learnable gating mechanism that modulates attention weights head-wise. Also, here queries, keys and values that are obtained via linear projections are transformed using rotary positional embeddings. The formed attention output vector is added to input by residual add operation.

After this the second normalization, LayerNorm2 is performed. The latent space $z^{(l)}$ is the output of this operation. Then $z^{(l)}$ is given to Feed Forward network, which consists of two linear transformations with Gaussian Error Linear Unit (GELU) activation function and output of this network is added to results of first Residual Add operation via second Residual Add operation. This operation forms output of the block, $h^{(l+1)}$, that is given to the input of next transformer block.

### 2.3.2 Latent statistics snapshot block

This block is aimed at collecting statistics of the model latent space $z^{(l)}$, formed after LayerNorm2 operation. This is the space that proposed model aims to regularize. Here, mean and covariance of $z^{(l)}$, approximated as a multivariate Gaussian distribution with diagonal covariation matrix, are extracted. These statistics are accumulated over each dataset training to obtain stable estimates for each domain.

During training stages, the previously computed statistics are used as reference distribution. The current latent distribution is compared to the reference one, obtained after previous domain training, via KL divergence

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

and this divergence, multiplied by scaling coefficient, is used as a regularization term in the models' loss function.

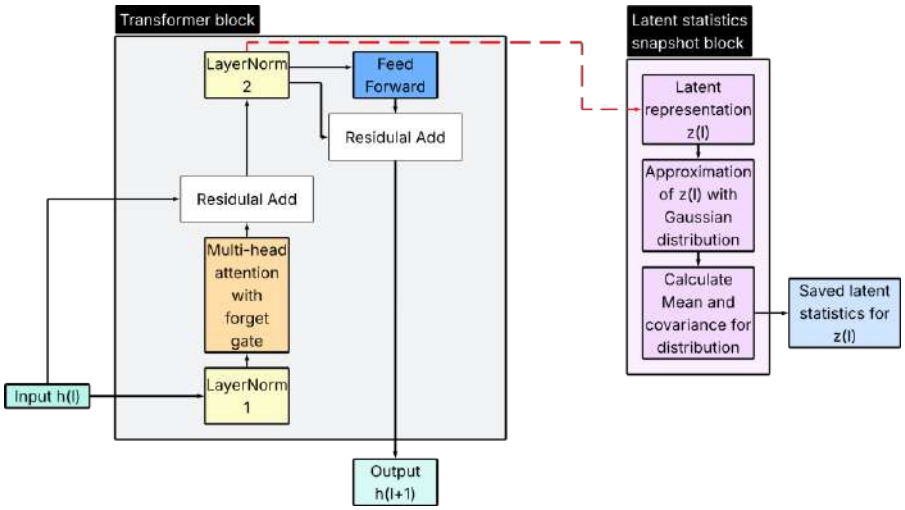


Fig. 1: Scheme of transformer block used in proposed approach

### 2.4 Experiment setup

In this study, the experimental setup, described below, is designed to evaluate the effectiveness of the proposed approach, compared to other existing approaches in controlled continual learning scenario with sequential domain shifts. This includes datasets selection and preprocessing, model configuration and hyperparameters, training procedure, experimental protocol and evaluation metrics.

#### 2.4.1 Data preparation

For evaluating the proposed model on the task of natural language processing (NLP), specifically autoregressive language modeling, in continual learning scenario, were selected 3 text corpora, that represent different domains: Wikipedia [18], CC-News [19], EURLEX [20].

These datasets differ in language style and semantic structure, and this makes them suitable for analyzing catastrophic forgetting in sequential domain adaptation. Also it creates realistic continual learning setup, where model is sequentially exposed to heterogeneous domains. This allowing to analyze how effective is proposed approach. Below is a brief description of each chosen dataset.

##### 2.4.1.1 Wikipedia

This dataset is derived from English version of Wikipedia. It contains encyclopedic articles with wide range of topics, such as science, technology, culture, history, etc. Each record in this dataset contains an independent text article. The articles in this dataset typically contain long paragraphs with clear structure and are written in formal descriptive language style. In original dataset in addition to the text article there are some metadata fields, such as title, section headers, and auxiliary info, but in this study, they are not used.

##### 2.4.1.2 CC-News (News)

This dataset contains news articles that are collected from a wide range of online media sources, covering different topics: economics, politics, technology and events. Each record of this dataset contains: the text of the article, title of this article, date of publication, source metadata, related image / images. Articles in this dataset contain slightly shorter paragraphs, compared to Wikipedia and are written in journalistic language style without clear structure, representing more live-use language constructions. For this study only text fields of this dataset that contain articles are used.

##### 2.4.1.3 EURLEX (Legal)

This dataset contains documents retrieved from European Union legal database, including regulations, directives, laws, etc. Each record of this dataset contains: legal document text, document identifiers, classification metadata. Texts in this domain are smaller than in Wikipedia and News, have complex sentence structures and are written in highly formal language style. Also this domain has a lot of specialized terminology and specific expressions, compared to Wikipedia and CC-News. For experiments in this study only legal document text is used.

All datasets that are used in experiments in this study are publicly available and were accessed through the HuggingFace Datasets data library for reproducibility.

#### 2.4.1.5 Motivation of datasets selection

Datasets described above were chosen to create a continual learning scenario with increasing domain shift. Such setup is essential for evaluating of robustness of the proposed approach under realistic conditions when model is exposed to heterogeneous data distributions.

For this task it is very important to take diverse domains, because when training on new corpora, the model needs to adapt to significantly different language distribution. On transition from Wikipedia to News, there are stylistic variations, structure shifts, and on transition from News to Legal there are substantial syntactic, lexical and structural differences.

Training order is the following:

$$\text{Wikipedia} \rightarrow \text{News} \rightarrow \text{Legal}$$

During training on new domain, model do not have access to previous domains data. Also, all models` parameters are training on new domains.

To maintain similar training conditions for each domain, each dataset was truncated to fixed token budget of 35 000 000 tokens. This number was taken, because this is the approximate size of the smallest dataset (Legal), obtained after preprocessing.

#### 2.4.1.6 Preprocessing

To maintain consistent training across domains and ensure that no noise in data is affecting results, the data preprocessing was applied to all used datasets. Each document was normalized by removing redundant whitespace and other formatting artifacts, and empty samples were removed too. Also, all text samples, containing less than 200 characters, are removed. Assuming that these datasets may contain not only English language samples, auto language detection was used and all non-English samples are removed. Also, documents are removed if they contain any hyperlinks, URLs or have low proportion (less than 0.6) of alphabetic characters (noisy or non-linguistic content). On each dataset, deduplication via MD5 hashing was made. Each domain dataset is preprocessed separately. All datasets were split in train, validation and test parts, train is 90%, validation is 5% and test is 5%. Tokenization parameters are shown in Table 1:

Table 1

Tokenization configuration

| Parameter name | Value |
|---|---|
| Tokenizer | GPT-2 |
| truncation | False |
| add_special_tokens | False |
| pad_token | eos_token |
| Padding strategy | Dynamic |
| Attention mask saving | False |

#### 2.4.1.4 Datasets summary

After all datasets undergone preprocessing, lets compare them visually to understand their structural differences.

First, distribution of document sizes across all domains. It can be clearly seen (Fig. 2) that for Wikipedia domain there are more documents with size of 10000+ tokens than News and Legal domains. Also, it is clear that in Legal domain all documents are smaller than 10000 tokens.

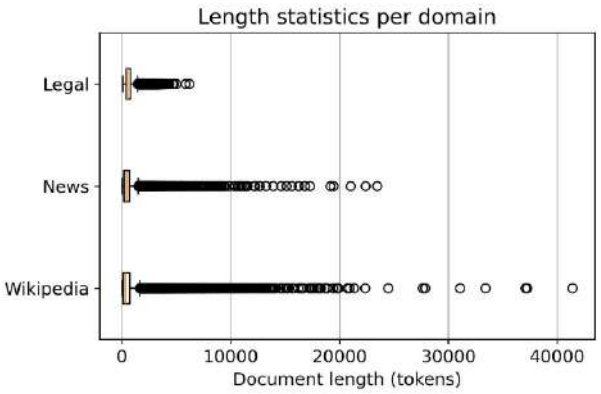


Fig. 2: Distribution of document sizes for all domains

Now about domain similarity. To show how much domains differ from each other, the domain similarity matrix was computed using TF-IDF representations and cosine similarity as a metric of semantic closeness.

The heatmap (Fig. 3) illustrates semantic similarity between all used datasets. As expected, difference between the Wikipedia and News is not so big, while difference between Legal domain and all other domains is way bigger.

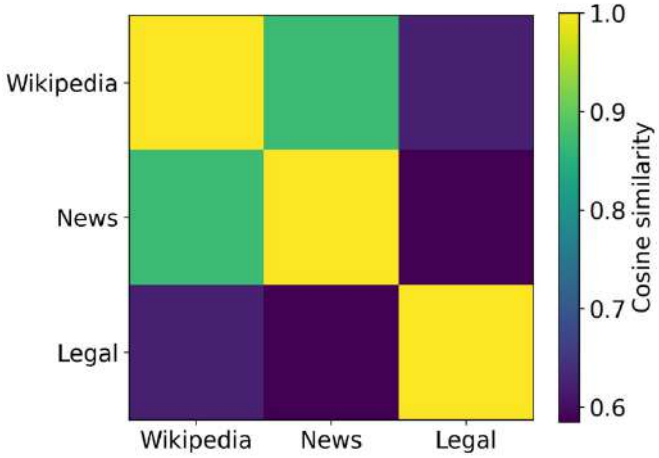


Fig. 3: Domain semantic similarity heatmap

Final datasets sizes (documents count and total amount of tokens that are used in all experiments) are shown in Table 2:

Table 2

Final datasets info

| Dataset name | Documents | Tokens |
|---|---|---|
| Wikipedia | 49911 | 34997875 |
| News | 63298 | 34999886 |
| Legal | 44961 | 32156942 |

### 2.4.2 Model architecture

For described task a custom casual transformer language model is used. Its architecture is designed to approximate mid-scale decoder-only architecture and on the other hand be computationally feasible on available equipment. All model parameters and configurations described in this section are frozen for all experiments that are done in this study.

Final model parameters that are used for all experiments are described in the Table 3:

Table 3

Model parameters

| Parameter name | Value |
|---|---|
| Number of layers | 6 |
| Hidden size | 512 |
| Number of attention heads | 8 |
| Attention head dimension | 64 |
| Feed-Forward Network dimension | 2048 |
| Vocabulary size | 50257 |
| Normalization type | LayerNorm |
| LayerNorm ε | 1e-5 |
| Affine parameters | True |
| Final LayerNorm | True |
| FFN activation function | GELU |
| LM Head | Linear |
| LM Head bias | False |
| Basic loss function | CrossEntropy |

For better model performance with long sequences, Rotary Positional embeddings [12] are used separately in each attention head. For $\vec{q}, \vec{k}$ vectors used in self-attention mechanism, they are transformed before dot-product using (22):

$$\mathrm{RoPE}(\vec{q},\vec{k}) = \left(\vec{q} * \cos(\theta) + \mathrm{R}(\vec{q}) * \sin(\theta), \vec{k} * \cos(\theta) + \mathrm{R}(\vec{k}) * \sin(\theta)\right), \quad (22)$$

where $\vec{q} = (q_1, q_2, \dots, q_n)$;

$$\mathrm{R}(\vec{q}) = \vec{q} * \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix};$$

$\theta = p * \omega_i$.

where p is token position;

$$\omega_i = 10000^{-\frac{2i}{d}}.$$

### 2.4.3 Training arguments

The training configuration, noted in Table 4 is used:

Table 4

Model training hyperparameters

| Hyperparameter name | Value |
|---|---|
| Optimizer | AdamW |
| Learning rate | 2e-4 |
| Batch size | 1 |
| Gradient accumulation steps | 16 |
| Effective batch size | 16 |
| Number of epochs per domain | 1 |
| Gradient clipping | 1.0 |
| Optimizer reinitialization | True |
| Max_sequence_length | 512 |
| Chunking | No |
| Random seed | 42 |

The hyperparameters described in Table 4 are frozen for all experiments that are conducted in this study.

Such hyperparameters were selected empirically, based on preliminary experiments and common practices in experiments with language models. Due to equipment technical limitations, automatic hyperparameter optimization methods were not employed. Chosen hyperparameters demonstrated stable performance on available equipment and therefore were retained for all experiments.

Final used mode size is 70378496 parameters.

### 2.4.4 Experiment list

A sequence of experiments is conducted in order to compare proposed model efficiency with existing approaches to continual learning task, that are not using any memory buffers (replay), for the fair comparison. Brief descriptions of each experimental setup, conducted in this study are provided in the Table 5.

Table 5

Experiments description

| Approach | Description |
|---|---|
| Naive fine-tuning | No regularization methods |
| Elastic Weight Consolidation (EWC) | Parameter regularization method based on Fisher information |
| Latent regularization with modified attention | Proposed model |
| LoRA | Parameter isolation method |

For comparison with LoRA the total adaptation budget was divided into three isolated domain-specific adapters. Each domain received a dedicated LoRA rank of 128. Such configuration eliminates parameter interference between domains. During domain training parameters that belong to other domains are frozen. This baseline represents a favorable setting for LoRA and is giving it good conditions for its continual learning performance under a fixed adaptation budget. The total LoRA rank was intentionally set to a relatively large value of 384, compared to commonly used LoRA configurations, reducing possibility of baseline performance constraining by small adapter size. Detailed LoRA configurations are stated in Table 6:

Table 6

LoRA configurations

| Parameter name | Value |
|---|---|
| r | 128 |
| lora_alpha | 256 |
| lora-dropout | 0.1 |
| target_modules | qkv |
| bias | none |
| task_type | None |

For proposed method $\lambda = 2 * 10^{-2}$ is defined. This parameter affects the amount of model punishment for information loss.

For EWC $\lambda = 1000$ is used. In EWC method, the regularization term is weighted by the Fisher Information Matrix, whose values are typically several orders of magnitude smaller than the task loss. Consequently, substantially larger $\lambda$ value is required to achieve a meaningful regularization effect.

### 2.4.5 Evaluation metrics

For model evaluation following metrics are used:
Perplexity (PPL) (23) is measured on validation set after each training stage and serves as primary metric for evaluating model performance.

$$PPL = e^{E[L]}, \quad (23)$$

where L is loss function for each evaluation batch;

E[L] is the loss average value on the validation data. Forgetting for domain $D_i$ after training on domain $D_j$ is defined as (24):

$$F_{i,j} = PPL_{after\ D_j}(D_i) - PPL_{after\ D_i}(D_i), \quad (24)$$

where $PPL_{after\ D_j}(D_i)$ is PPL on domain $D_i$ after training on domain $D_j$;

$PPL_{after\ D_i}(D_i)$ is perplexity on domain $D_i$ after training on this domain.

## 3. Results and case study

This section presents experimental evaluation of the proposed approach and evaluation of its effectiveness compared to other existing approaches described in Table 4. The results include both quantitative and qualitative analysis of model behavior across training stages. Here are presented perplexity values, domain-wise forgetting measures and stability-plasticity trade-off analysis. In addition to the standard evaluation metrics, in this section the internal model dynamics is examined via analysis of latent representation drift across model layers. Also here computation budget is

The case study is based on experimental setup described in section 2.4 and focuses on the autoregressive language modeling under a continual learning scenario with three domain shifts. All metrics that are stated in the section 2.4.5 are computed for the case study task equally for all experimental approaches, stated in Table 5.

### 3.1 Perplexity curves

These plots are showing perplexity values after training on each domain: perplexity on Wikipedia after training on News, perplexity on News after training on Legal and perplexity on Wikipedia after training on Legal. These graphics show how model adapted to new domain. The lower is perplexity on domain the better adaptation of model to this domain. This is the measure of model plasticity.

Fig.4 represents proposed model.

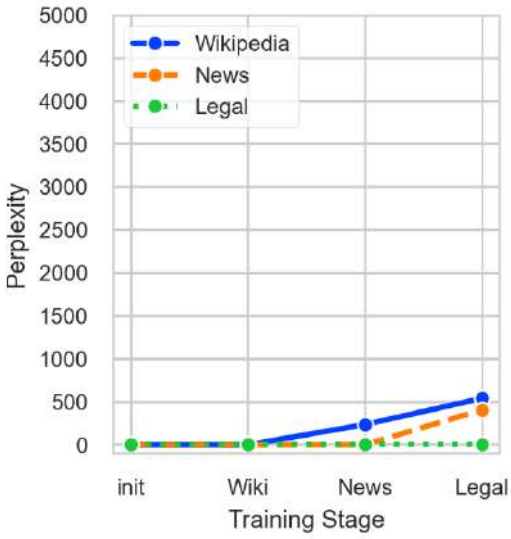


Fig. 4: Per-domain perplexity dynamics of proposed model

Fig. 5 corresponds to the model with EWC regularization.

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

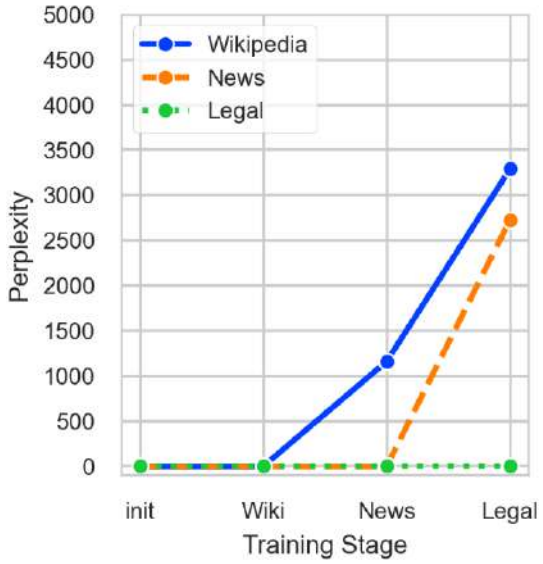


Fig. 5: Per-domain perplexity dynamics of model with EWC regularization

Fig. 6 represents model with naive fine-tuning.

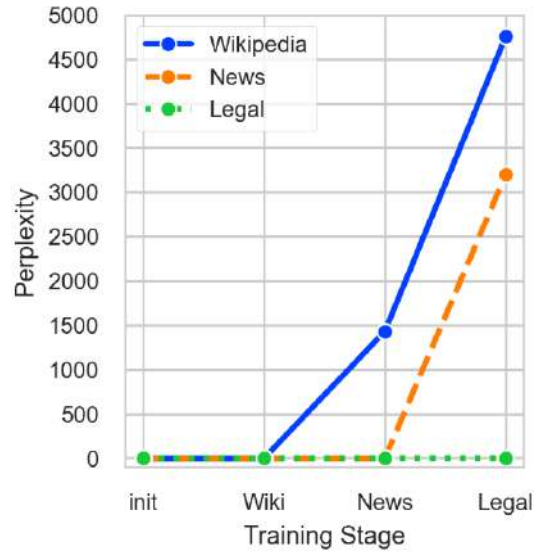


Fig. 6: Per-domain perplexity dynamics of model with naive fine-tuning

Figure 7 shows the perplexity dynamics for model with LoRA.

Here, the stability of perplexity curve for domain after learning on it shows the stability of such trained adapter.

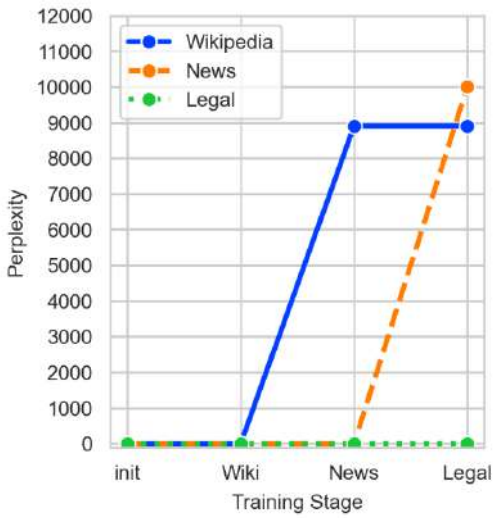


Fig. 7: Per-domain perplexity dynamics of model with LoRA

### 3.2 Forgetting curves

These plots show the dynamics of model productivity degradation on previously learned domains after learning on new domain. The lower is forgetting the more stable is model. This is a measure of model stability: ability to preserve previously acquired knowledge.

These plots help to understand how using different approaches to continual learning (Table 3) influences the amount of forgetting.

Fig. 8 represents proposed model.

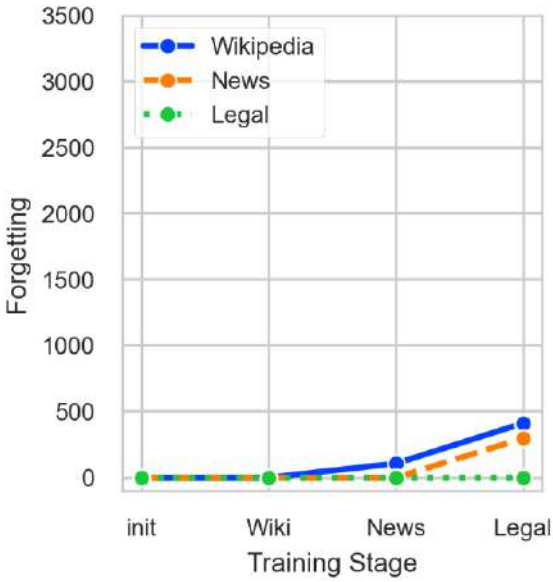


Fig. 8: Per-domain forgetting dynamics of proposed model

Fig. 9 corresponds to the model with EWC regularization.

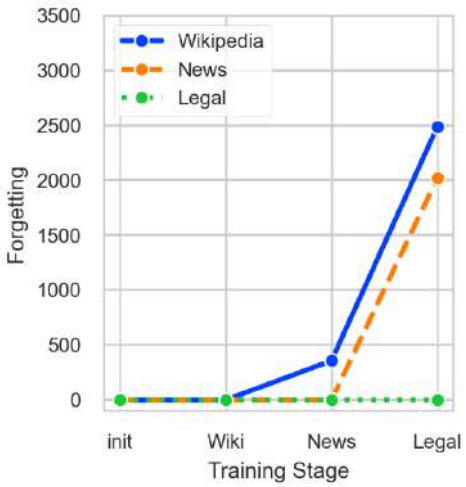


Fig. 9: Per-domain forgetting dynamics of model with EWC regularization

Fig. 10 illustrates the behavior of model with naive fine-tuning.

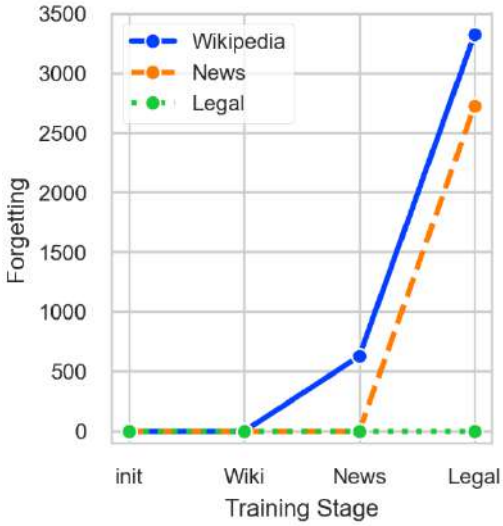


Fig. 10: Per-domain forgetting dynamics of model with naive fine-tuning

Figure 11 shows forgetting dynamics of model with LoRA.

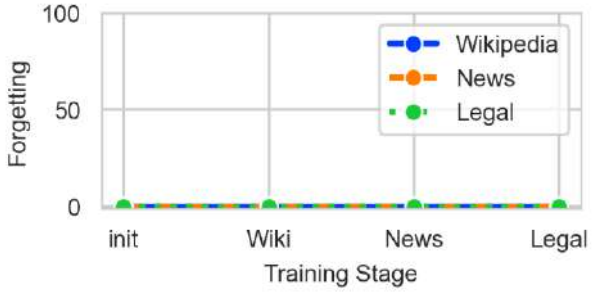


Fig. 11: Per-domain forgetting dynamics of model with LoRA

### 3.3 Stability-plasticity trade-off

These plots sow how model stability and plasticity are connected. In these plots plasticity is measured as the perplexity on the newly learned domain after model is trained on it, reflecting how well the model adapted to new data. Stability is calculated as an average forgetting on previously learned domains after model is trained on the new domain. Here, colored region on plots is stability-plasticity trade-off. The smaller this region is, the better the model maintains the optimal balance between adaptation and previous knowledge retaining. The numerical values on these plots are computed area of this region. Fig. 12 represents proposed model.

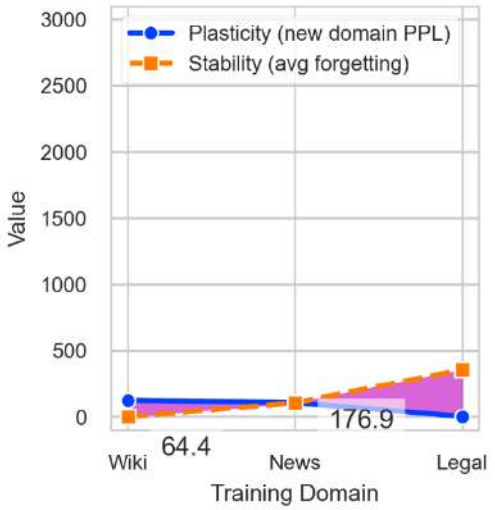


Fig. 12: Stability-plasticity trade-off dynamics of proposed model

Fig. 13 corresponds to the model with EWC regularization.

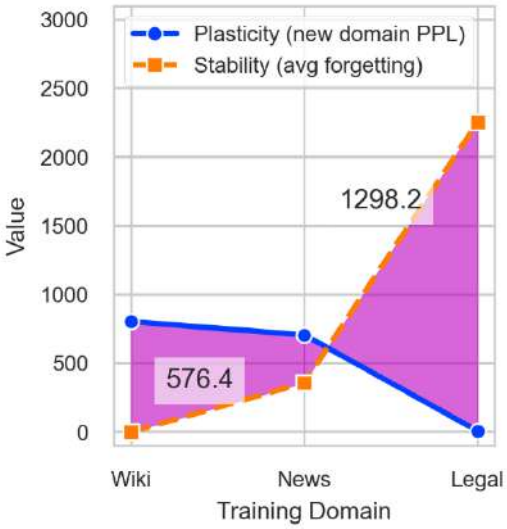


Fig. 13: Stability-plasticity trade-off dynamics of model with EWC regularization

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

Fig. 14 illustrates the behavior of model with naive fine-tuning.

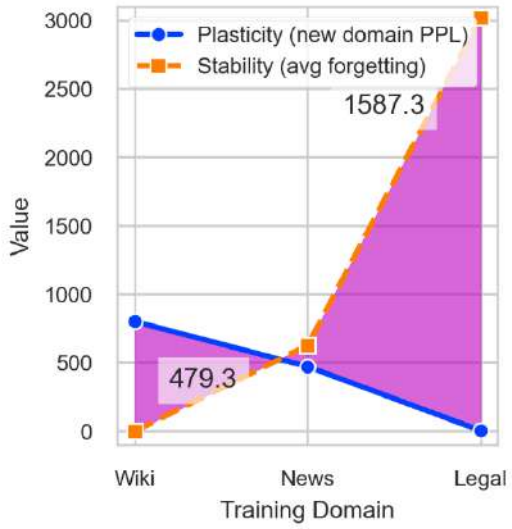


Fig. 14: Stability-plasticity trade-off dynamics of model with naive fine-tuning

Figure 15 represents stability-plasticity trade-off for model with LoRA.

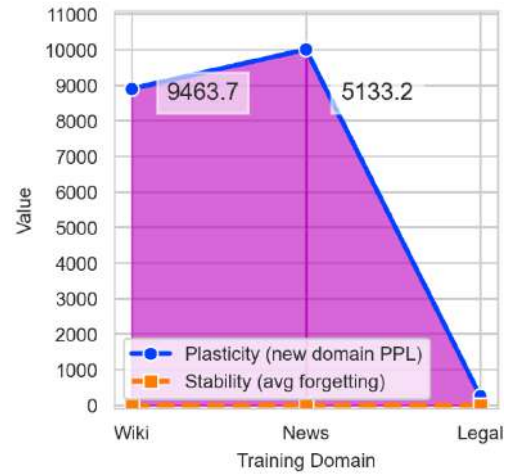


Fig. 15: Stability-plasticity trade-off dynamics of model with LoRA

### 3.4 Layer-wise representation drift heatmap

The layer-wise representation drift heatmap visualizes the changes in latent representations across transformer layers during sequential domain adaptation. Only the layers to which the proposed regularization is applied are considered in all experiments. Latent representations for this analysis are extracted after the second Layer Normalization operation, as shown in the model layer architecture (Fig. 1).

Each cell in the heatmap is the distance between latent mean vectors of a given layer before and after domain transition. To compute this value, for each layer hidden representations are aggregated by averaging across all tokens and samples of the dataset. The resulting mean vectors for two consecutive training stages are compared via Euclidean distance (L2). Here, L2 distance is chosen because it allows to estimate the distance between mean vectors of latent representations in their Gaussian approximations. The Eucledian distance between them is a component of KL divergence between two Gaussian-like distributions. This provides the quantitative assessment of representation drift and the regularization term used in proposed model. This provides a measure of representation drift on each layer.

Here larger values are interpreted as stronger changes in the latent representations, which is a sign of the reorganization of the model internal knowledge. Such reorganization is associated with adaptation to new domains, but also implies the loss of distortion of previously learned information.

Smaller distances can be interpreted as the relative stability of the latent representations showing that the previously acquired knowledge is preserved. Such visualization provides insight into the role of different layers in adaptation process and highlights, where knowledge reorganization is the largest, and where it is the smallest and shows how stable the hidden representations are during training of the proposed model and of the models with different approaches.

Fig. 16 presents the representation drift heatmap for the proposed model.

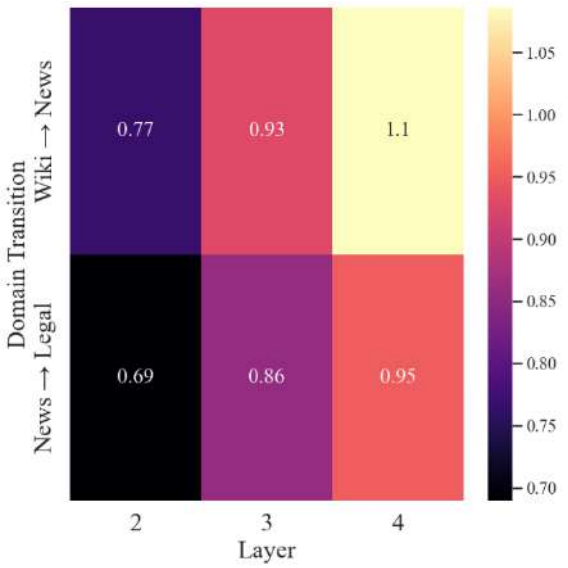


Fig 16: Layer-wise representation drift heatmap of proposed model

Fig. 17 corresponds to the heatmap for model with EWC parameter regularization.

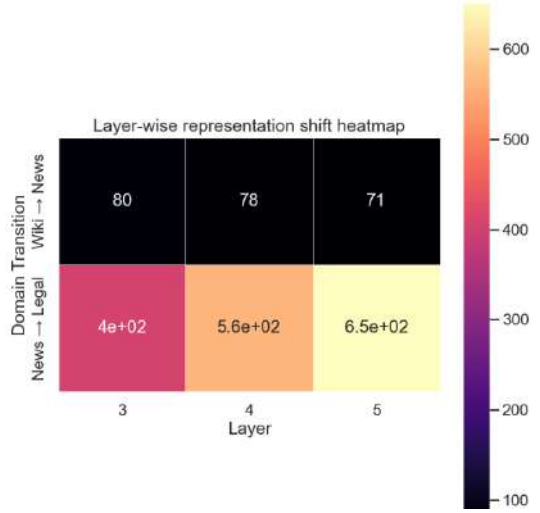


Fig. 17: Layer-wise representation drift heatmap of model with EWC regularization

Fig. 18 shows the results for the model with naive fine-tuning.

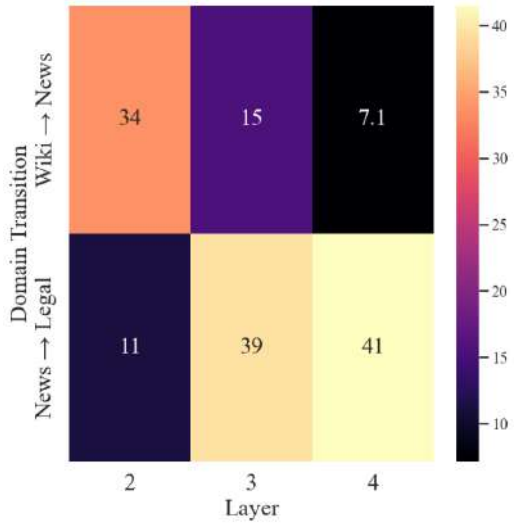


Fig. 18: Layer-wise representation drift heatmap of model with naive fine-tuning

### 3.5 Result tables

All experiments were conducted on a single NVIDIA RTX 3060 GPU.

In addition to isolated LoRA adapters, another experiment was conducted, using only one isolated adapter for all domains (all LoRA parameters for this experiment are stated in Table 6). In the Table 7 forgetting after training on domains is shown.

Table 7

Forgetting of the model after training on the defined domains

| Approach | Forgetting | | |
|---|---|---|---|
| | Wikipedia after News | News after Legal | Wikipedia after Legal |
| Naive fine-tuning | 629.161 | 2725.362 | 3324.745 |
| EWC parameter regularization | 357.2 | 2020.09 | 2130.8 |
| Proposed model | 107.397 | 297.960 | 306.086 |
| LoRA (3 adapters) | 0 | 0 | 0 |
| LoRA (1 adapter) | 299.885 | 8487.849 | 6593.650 |

In Table 8 perplexity after training on domains is stated.

Table 8

Final perplexity measure after model training on all domains

| Approach | Domain | | |
|---|---|---|---|
| | Wikipedia | News | Legal |
| Naive fine-tuning | 4756.834 | 3198.918 | 6.018 |
| EWC parameter regularization | 3291.918 | 2727.105 | 7.898 |
| Proposed model | 540.632 | 406.985 | 3.536 |
| LoRA (3 adapters) | 9008.937 | 10561.078 | 269.496 |
| LoRA (1 adapter) | 21521.849 | 21782.651 | 1272.227 |

The computation budget for this experiments are shown in Table 9:

Table 9

Computation budget

| Approach | GPU-hours | Relative Cost |
|---|---|---|
| Naive fine-tuning | 1.845 | 100% |
| Proposed model | 1.755 | 95.1% |
| EWC parameter regularization | 2.145 | 116.3% |
| LoRA (3 adapters) | 1.314 | 71.2% |
| LoRA (1 adapter) | 1.284 | 69.6% |

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

Also, to ensure robustness of the proposed method to random seed variation and its stability, 5 independent runs were conducted, using new random seed every time.

Detailed results with mean, standard deviation and 95% confidence interval values are stated in Table 10:

Table 10

Proposed method stability test results

| Metric | Mean value | Standard deviation | 95% confidence interval |
|---|---|---|---|
| Final PPL (Legal) | 3.55 | ± 0.03 | ± 0.04 |
| Final PPL (News) | 399.16 | ± 7.04 | ± 8.74 |
| Final PPL (Wikipedia) | 533.38 | ± 24.91 | ± 30.9 |
| Forgetting (Wikipedia) | 407.18 | ± 24.87 | ± 30.88 |
| Forgetting (News) | 291.27 | ± 6.07 | ± 7.54 |

### 3.6 Comparison with LoRA

Having separate parameters for each domain guarantees absence of scenarios, when parameters for certain domain are changed during adaptation to another domain, which results in absence of forgetting. But, even in the favorable scenario for LoRA (3 adapters), the quality of model adaptation is significantly lower (higher PPL), than for the proposed method (Table 8). Also, in the LoRA (3 adapters) configuration used in this study, further adaptation to other domains is impossible without increasing model parameter count via addition of new adapters.

On the other hand, proposed method does not have such limitations. Stability-plasticity balance (Fig. 15), as expected, is worse than any other method, tested in this study (Fig 14, Fig. 13, Fig. 12). In addition, separate isolated adapters for each domain makes knowledge transfer between domains impossible. For LoRA (1 shared adapter for all domains), situation is even worse than for 3 isolated adapters. This method achieved the worst PPL among all tested (Table 8). Also, the average forgetting for LoRA (1 adapter) is the highest in comparison to other tested methods.

### 3.7 Discussion

For naive fine-tuning model (Fig. 6), perplexity is very high on first two domains and is significantly lower on the third. This means that the model adapted to new domains by losing knowledge on previous domains. In EWC model (Fig. 5), there are some improvements on stability on Wikipedia and News, but by looking at Table 8 it can be concluded that the EWC model has poorer plasticity on the Legal domain, compared to naive fine-tuning model. For the proposed model (Fig. 4), perplexity on all domains is the lowest compared to other tested models, and as a result has the highest plasticity. For naive fine-tuning (Fig. 10), forgetting after each new domain increases drastically and cumulative forgetting on Wikipedia is the highest, compared to other models (Fig. 8), (Fig. 9). Such model behavior means that the model almost completely overwrites previously learned knowledge when trained on a new domain. Here the classic catastrophic forgetting phenomenon occurs, demonstrating the limitations of simple sequential fine-tuning without the use of any additional mechanisms for knowledge preservation. For model with EWC parameter regularization (Fig. 9), forgetting is lower, compared to naive fine-tuning, but not substantially lower. Parameter updating constraint here prevents some knowledge loss, but only partially. Although parameter regularization slows down degradation, the cumulative forgetting is still high after introduction of the new domain. On the other hand, in the proposed model (Fig. 8), forgetting is significantly smaller than in EWC and naive fine-tuning models, cumulative increase of forgetting is also almost absent, which indicates that this model is able to maintain previously learned knowledge during adaptation to new domains. As a result, the proposed model demonstrates higher stability compared to the naive fine-tuning and EWC parameter regularization.

In quantitative terms proposed model reduces forgetting by approximately 11 times, compared to naive fine-tuning and by approximately 7 times compared to the model with EWC regularization. These results confirm the effectiveness of the proposed approach in reducing forgetting. For the naive fine-tuning model (Fig. 14), the trade-off region is the largest among all tested models, meaning that it has the poorest stability-plasticity balance. Next, the EWC model (Fig. 13) has better overall stability-plasticity balance compared to the naive fine-tuning. However, the improvement is still limited. During the first domain transition the trade-off is slightly worse than in the naive fine-tuning, highlighting that parameter regularization can reduce adaptability to the new data. The stability-plasticity balance for the proposed model (Fig. 12), is the best among all tested models as it has the smallest area between curves. This shows that the proposed model can simultaneously achieve low forgetting on the previous domains, while maintaining good performance on new domains. As a conclusion, proposed model effectively minimizes forgetting and at the same time maintains good adaptability to new domains. This

result shows that the architectural modifications proposed in this study help in regulating model learning dynamics and improving of stability-plasticity balance, which is one of the key challenges in continual learning [1]. For the naive fine-tuning model (Fig. 18), representation drift is significant on all layers, especially on transition from News to Legal domain. Model reorganizes its internal representations to adapt to the new domain, but this reorganization leads to the loss of previous knowledge. For the EWC model (Fig. 17), representation drift is even larger than in naive fine-tuning model. Such results may seem counterintuitive but show one important observation: stability of the model parameters alone does not guarantee the stability of the representations and knowledge. Therefore, standalone parameter regularization may not be sufficient for knowledge retention.

For the proposed model (Fig. 16), representation drift is the smallest among all tested models. This shows that the latent representations remain significantly more stable than in the naive fine-tuning and EWC model. It proves that proposed regularization has a significant effect on latent space stability and prevents excessive reorganization of knowledge (small reorganization is needed though, as it represents adaptation to the new domain). This result shows that the proposed approach effectively prevents forgetting on the representation level.

Proposed method also maintains high stability in repeated runs with random seed variation (Table 10). The resulting 95% confidence intervals are very narrow compared to the metric values, confirming high statistical significance of the proposed method.

Besides continual learning quality metrics, the computational efficiency of the proposed method was evaluated using the computation budget, measured in GPU-hours required for model training. This metric reflects the practical computational cost of the proposed approach and enables a direct comparison with alternative continual learning methods. As shown in Table 9, the proposed model requires fewer GPU-hours than naive fine-tuning and EWC parameter regularization, demonstrating its computational efficiency while maintaining superior knowledge retention. Only LoRA requires fewer GPU-hours due to the significantly smaller number of trainable parameters. However, this reduction comes at the cost of lower adaptation capability and the limitations discussed in Section 3.6.

## 4. Conclusions

In modern days there is a growing need of LLM-based AI systems (AI agents), which will perform the role of, for example, a personal work assistant, a company advisor, etc. In such conditions these agents must be capable of operation in conditions of constant, lifelong learning on new incoming information and adaptation to new tasks. This new information and tasks can be very different to what model already know, so to maintain an acceptable level of quality such systems must retain their previous knowledge while effectively adapting to new information, otherwise the problem of catastrophic forgetting will arise.

To address this problem in transformer-based language models that are operating in continual learning scenarios with sequential domain shifts, a representation-level regularization framework inspired by Bayesian uncertainty with FoX-inspired forgetting attention mechanism is proposed in this paper.

The proposed method stabilizes latent representations within transformer layers both directly by modeling them as Gaussian distributions and minimizing the KL divergence between distributions, obtained during training steps, and indirectly, by regulating contextual information flow from attention mechanism. Results obtained in this paper demonstrate that the proposed approach significantly outperforms naive sequential fine-tuning and Elastic Weight Consolidation parameter regularization method and two LoRA configurations in terms of stability (knowledge retention) and adaptability to new domains.

The main contribution of the study is the proposed representation-level continual learning framework, which provides an integrated improvement in continual learning performance by simultaneously reducing forgetting by up to elevenfold relative to naive fine-tuning (sevenfold relative to EWC), achieving the lowest perplexity on previously learned domains, improving the stability–plasticity balance, and maintaining competitive computational efficiency without relying on replay memory. The experimental results confirm that the objective of this study has been achieved.

The proposed latent representation modeling relies on diagonal Gaussian approximation of the latent feature distributions. This assumption makes the proposed method computationally feasible in high-dimensional latent representation spaces, but it ignores the potentially possible correlation between latent dimensions, which can possibly lead to poorer regularization. Capturing these correlations with use of richer probabilistic models could potentially improve representation stability, but with introduction of additional computation overhead. This is the entry for further studies.

Unlike meta-learning approaches, the proposed method is not tied to a specific training paradigm. It can be incorporated into the optimization process of transformer language models independently of the underlying training strategy. In contrast, meta-learning defines a learning paradigm aimed at optimizing model adaptation to new tasks through a dedicated meta-training procedure [21]. Since these approaches address different aspects of

the learning process, they should be viewed as complementary rather than competing.

Consequently, the proposed regularization can, in principle, be integrated with meta-learning frameworks to simultaneously improve adaptation capability and knowledge retention, which is a promising direction for future research.

From a practical perspective, the proposed method can be applied to transformer-based language models that require continual adaptation to evolving data while preserving previously acquired knowledge. As an example, it can be used in conversational AI systems, intelligent virtual assistants, domain-specific language models, decision support systems, and other AI services that are continuously updated with new information. Since LatReg operates as a representation-level regularization technique without requiring replay memory or modifications to the overall transformer architecture, it can be incorporated into existing continual learning pipelines with minimal changes to the training procedure.

**Contributions of authors:** General structure of the work, research idea, comparison and drafting of manuscript – **Arsentii Kriukov**; research led with overall supervision, revision – **Yurii Pushkarenko**.

All authors have read and approved the final manuscript for publication.

## Conflict of Interests

The authors declare that they have no conflict of interest in relation to this research, whether financial, personal, authorship or otherwise, that could affect the research and its results presented in this paper.

## Financing

The research was conducted without financial support.

## Data Availability

The implementation of datasets preprocessing and proposed method can be accessed via GitHub https://github.com/Whookk/LatReg.

## Use of Artificial Intelligence

Artificial intelligence methods were used for literature search while creating the presented work. All the authors have read and agreed to the published version of this manuscript.

## References

1. Luo, Y., Yang, Z., Meng, F., Li, Y., Zhou, J., & Zhang, Y. An Empirical Study of Catastrophic Forgetting in Large Language Models During Continual Fine-Tuning. *IEEE Transactions on Audio, Speech and Language Processing*, 2025, vol. 33, pp. 3776–3786. DOI: 10.1109/taslpro.2025.3606231.

2. Jiang, M., Fan, J., & Li, F. Advances in continual learning: A comprehensive review. *Expert Systems with Applications*, 2025, vol. 294, article no. 128739. DOI: 10.1016/j.eswa.2025.128739.

3. Shin, H., Lee, J. K., Kim, J., & Kim, J. Continual Learning with Deep Generative Replay. *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS 2017)*, 2017, vol. 30, pp. 2990–2999. DOI: 10.48550/arXiv.1705.08690.

4. He, J., Guo, H., Zhu, K., Zhao, Z., Tang, M., & Wang, J. SEEKR: Selective Attention-Guided Knowledge Retention for Continual Learning of Large Language Models. *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 2024, pp. 3254–3266. DOI: 10.18653/v1/2024.emnlp-main.190.

5. Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., Hassabis, D., Clopath, C., Kumaran, D., & Hadsell, R. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 2017, vol. 114, no. 13, pp. 3521–3526. DOI: 10.1073/pnas.1611835114.

6. Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., Laroussilhe, Q. D., Gesmundo, A., Attariyan, M., & Gelly, S. Parameter-Efficient Transfer Learning for NLP. *Proceedings of the 36th International Conference on Machine Learning*, 2019, vol. 97, pp. 2790–2799. DOI: 10.48550/arXiv.1902.00751.

7. Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., & Chen, W. LoRA: Low-Rank Adaptation of Large Language Models. *ICLR 2022 Conference Paper*, 2022. Available at: https://openreview.net/forum?id=nZeVKeeFYf9 (accessed 18 March 2026).

8. Li, H., Tan, Z., Li, X., & Huang, W. ATLAS: Adapter-Based Multi-Modal Continual Learning with a Two-Stage Learning Strategy. *arXiv preprint*, arXiv:2410.10923, 2024. Available at: https://arxiv.org/abs/2410.10923 (accessed 18 March 2026).

9. Hu, G., Zhang, W., Ding, H., & Zhu, W. Gradient Episodic Memory with a Soft Constraint for Continual Learning. *arXiv preprint*, arXiv:2011.07801, 2020. Available at: https://arxiv.org/abs/2011.07801 (accessed 18 March 2026).

10. Buzzega, P., Boschini, M., Porrello, A., Abati, D., & Calderara, S. Dark Experience for General Continual Learning: a Strong, Simple Baseline. *Proceedings of the 34th International Conference on Neural Information Processing Systems*, 2020, article no. 1335. DOI: 10.48550/arXiv.2004.07211.

11. Nguyen, C. V., Li, Y., Bui, T. D., & Turner, R. E. Variational Continual Learning. *ICLR 2018 Conference Paper*, 2018. Available at: https://openreview.net/forum?id=BkQqq0gRb (accessed 18 March 2026).

12. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Gomez, A., Kaiser, Ł., & Polosukhin, I. Attention Is All You Need. *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, vol. 30, pp. 6000–6010. DOI: 10.48550/arXiv.1706.03762.

13. Bonnet, D., Cottart, K., Hirtzlin, T., Januel, T., Dalgaty, T., Vianello, E., & Querlioz, D. Bayesian continual learning and forgetting in neural networks. *Nature Communications*, 2025, vol. 16, article no. 9614. DOI: 10.1038/s41467-025-64601-w.

14. Cover, T. M., & Thomas, J. A. Elements of Information Theory. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2005, 748 p. DOI: 10.1002/047174882x.

15. Kingma, D. P., & Welling, M. An Introduction to Variational Autoencoders. *Foundations and Trends in Machine Learning*, 2019, vol. 12, no. 4, pp. 307–392. DOI: 10.1561/2200000056.

16. Lin, Z., Nikishin, E., He, X. O., & Courville, A. Forgetting Transformer: Softmax Attention with a Forget Gate. *ICLR 2024 Conference Paper*, 2024. Available at: https://openreview.net/forum?id=q2Lnyegkr8 (accessed 18 March 2026).

17. Xiong, R., Yang, Y., He, D., Zheng, K., Zheng, S., Xing, C., Zhang, H., Lan, Y., Wang, L., & Liu, T. On Layer Normalization in the Transformer Architecture. *Proceedings of the 37th International Conference on Machine Learning*, 2020, vol. 119, pp. 10524–10533. DOI: 10.48550/arXiv.2002.04745.

18. Wikimedia Foundation. Wikipedia dataset. *Hugging Face Datasets*, 2022. Available at: https://huggingface.co/datasets/wikimedia/wikipedia (accessed 18 March 2026).

19. Blagojevic, V. CC-News dataset. *Hugging Face Datasets*, 2020. Available at: https://huggingface.co/datasets/vblagoje/cc_news (accessed 18 March 2026).

20. Lesci, P. EURLEX-57K dataset. *Hugging Face Datasets*, 2021. Available at: https://huggingface.co/datasets/pietrolesci/eurlex-57k (accessed 18 March 2026).

21. Hospedales, T. M., Antoniou, A., Micaelli, P., & Storkey, A. J. Meta-Learning in Neural Networks: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022, vol. 44, no. 9, pp. 5149–5169. DOI: 10.1109/TPAMI.2021.3079209.

*Received 18.03.2026, Received in revised form 03.06.2026*
*Accepted date 07.07.2026, Published date 31.07.2026*

## АДАПТИВНА РЕГУЛЯРИЗАЦІЯ ЗНАНЬ ДЛЯ ТРАНСФОРМЕРНИХ АРХІТЕКТУР У ЗАДАЧАХ ПОСЛІДОВНОГО НАВЧАННЯ

***Крюков А.В., Пушкаренко Ю.В.***

*Статтю присвячено розробці механізму регуляризації латентних представлень у моделі на основі архітектури Transformer в умовах безперервного навчання за послідовної зміни доменів. Сучасні мовні моделі досягають високої якості у статичних сценаріях навчання, але вони залишаються обмеженими у випадках довготривалого функціонування, поступової адаптації до нових доменів і не є достатньо стійкими до забування, особливо за відсутності доступу до раніше спостережуваних даних. У цій статті досліджується можливість подолати зазначені обмеження шляхом поєднання методу, що ґрунтується на байєсівській невизначеності і механізмів забування, вбудованих у механізм уваги в єдиній трансформерній архітектурі. Метою дослідження є проєктування, реалізація та валідація трансформерної архітектури з багаторівневим механізмом регуляризації представлень, який забезпечує можливість мовним моделям на основі трансформерної архітектури ефективно адаптуватися до нового розподілу даних, зберігаючи при цьому раніше набуті знання. Запропонований підхід забезпечує досягнення рівноваги між адаптивністю моделі та її стабільністю в процесі безперервного навчання без потреби у повному перенавчанні моделі або збереженні старих даних. Для досягнення поставленої мети необхідно розв'язати такі завдання: формалізувати метод регуляризації латентних представлень, що поєднує регуляризацію латентних представлень моделі, засновану на байєсівській невизначеності, із механізмом масштабування уваги на рівні окремих голів механізму уваги; впровадження цього методу у трансформерну модель; розроблення експериментального сценарію безперервного моделювання мови з послідовними змінами домену; кількісне оцінювання забування моделі та компромісу між стабільністю та пластичністю; порівняння запропонованої моделі з класичним донавчанням без регуляризації, методом LoRA та методами регуляризації параметрів. Висновки показують, що запропонований метод забезпечує менший рівень забування, нижчу перплексію на раніше вивчених доменах та кращий компроміс між стабільністю та пластичністю порівняно з наївним донавчанням, LoRA та Elastic Weight Consolidation, за порівнянних обчислювальних витрат. Наукова новизна запропонованого підходу полягає в розробці селективної за шарами латентної регуляризації для безперервного моделювання мови, яка інтегрує механізм уваги із механізмом забування зі збереженням інваріантних до домену представлень через статистичне вирівнювання в латентному просторі для зменшення ефекту забування у сценаріях послідовного донавчання мовних моделей. На відміну від існуючих підходів до безперервного навчання, які розглядають регуляризацію моделі на рівні параметрів або пам'яті (з використанням попередніх даних), запропонований підхід здійснює регуляризацію на рівні латентних представлень, де використовує як пряму регуляризацію на основі невизначеності, так і непряму регуляризацію через модифікований механізм уваги, забезпечуючи стале безперервне моделювання мови в нестаціонарних середовищах.*

**Ключові слова:** *безперервне навчання; катастрофічне забування; регуляризація латентного простору; модифікація уваги; невизначеність; мовні моделі; обробка природної мови; трансформери; інформаційна ентропія.*

**Крюков Арсентій Володимирович** – магістр, кафедра теорії та технології програмування, Київський національний університет імені Тараса Шевченка, Київ, Україна,

**Пушкаренко Юрій Валерійович** – асистент, доктор філософії, кафедра теорії та технології програмування, Київський національний університет імені Тараса Шевченка, Київ, Україна

**Arsentii Kriukov** – Master's student; Department of Theory and Technology of Programming, Taras Shevchenko National University of Kyiv, Kyiv, Ukraine,
e-mail: kriukov_a@knu.ua. ORCID: 0009-0000-2443-9171.

**Yurii Pushkarenko** – Assistant Professor, Doctor of Philosophy; Department of Theory and Technology of Programming, Taras Shevchenko National University of Kyiv, Kyiv, Ukraine,
e-mail: yurii.pushkarenko@gmail.com, ORCID: 0009-0007-2560-2971.