CAL-ATS: Cosine Adaptive Lookahead Optimization for Automated Applicant Tracking Systems

Alla Abhiram[1], G.S. Abhinav[1], Nenavathu Pranay[1], Kislay Dilliwar[1], and Vijaya J[2]

[1] UG Student, DSAI/ECE, International Institute of Information Technology, Naya Raipur, India
[2] Assistant Professor, DSAI, International Institute of Information Technology, Naya Raipur, India
alla23102@iiitnr.edu.in, gudipudi23102@iiitnr.edu.in, nenavathu23101@iiitnr.edu.in, kislay23102@iiitnr.edu.in, vijaya@iiitnr.edu.in

Abstract. Recruitment systems require Automated Applicant Tracking Systems (ATS) to handle their current recruitment needs yet these systems face major problems because their results remain difficult to understand. Deep learning technology increases screening accuracy however traditional models function as "black boxes" which reduce trust from recruiters. The researchers developed a Dual-Stream Transformer system which predicts hiring decisions while producing natural language explanations for its predictions. The joint discriminative-generative optimization process experiences problems because its two elements create incompatible gradient updates. The research team developed **Cosine-Adaptive Lookahead (CAL)** which acts as a new optimization tool that adjusts weight synchronization according to how closely gradient paths match each other. The framework we developed shows through testing on 10,175 resume-job pairs that CAL effectively tackles task interference which leads to faster results and better performance than both standard AdamW and fixed-interpolation testing methods. Our results provide a principled path toward more stable and transparent multi-task learning in high-stakes NLP applications.

Keywords: Multi-Task Learning · Optimization · Explainable AI · NLP · CAL

1 Introduction

The current process of hiring workers today experiences an extraordinary increase in online job applications. Large companies encounter difficulties because they receive thousands of job applications when they open a single position which creates a situation that human recruiters cannot handle through their manual work. Organizations now need to use Applicant Tracking Systems (ATS) because these systems have become essential for their hiring process. The initial

version of these systems required users to depend on strict word identification methods together with regular expression (regex) technology. The methods show high computational speed but they suffer from major weaknesses because they work incorrectly when they encounter unknown words and they cannot differentiate between people who possess basic Python skills and those who have written books about Python [1-2].

The economic consequences of this semantic distinction between two meanings carry substantial weight. The cost of a "bad hire" is estimated to be upwards of 30% of the employee's first-year earnings but the opportunity cost of rejecting a *star* candidate due to algorithmic rigidity is often even higher. Recent advancements in Natural Language Processing (NLP) especially the emergence of Transformer-based architectures like BERT have promised a solution. The models develop high-dimensional vector space representations of text which enable them to understand the complex relationship between a candidate's multiple work experiences and the specific job requirements [3-4].

The Black Box Problem

The use of these models in high-stakes decision-making processes leads to a major challenge which requires its solution through the process of **Explainability**. The GDPR's "Right to Explanation" regulation and similar rules in multiple jurisdictions require that automated decision systems need to provide interpretable output. The deep neural network produces a "black box" effect through its basic "Select" and "Reject" output, which results in decreased trust from recruiters, while it introduces ethical issues about algorithmic bias. A model needs to explain its reasons for rejecting a candidate by describing two possibilities: the candidate lacks required domain expertise or their experience level does not meet the position's requirements [5-6].

Researchers have adopted Multi-Task Learning (MTL) as a solution which enables them to create development models that simultaneously perform candidate evaluation and produce natural language explanations. Discriminative tasks which include classification and generative tasks which involve text generation face major challenges during their joint optimization process. The generation of complex text requires gradient updates which disrupt the stable conditions needed to achieve precise classification results, according to the described phenomenon of *gradient interference* which Yu 2020 [7] describes. The first-order optimization methods which include AdamW face difficulties when they deal with loss functions that have conflicting elements because this situation results in suboptimal results, where one task takes precedence over others (task dominance) or where shared representation stops working (catastrophic forgetting) [8]. In this paper, we propose a novel approach to stabilize multi-task resume screening. We argue that standard optimization trajectories are insufficient for the dual-objective nature of modern ATS. The contributions of this work are as follows:

– **Cosine-Adaptive Lookahead (CAL) Optimizer:** We present a weight synchronization optimization method which uses gradient alignment to de-

termine the optimal synchronization times. The CAL system uses task gradients' geometric relationship through cosine similarity to determine its weight update intervals between "Fast" and "Slow" updates.
- **Dual-Stream Architecture:** Our hybrid architecture combines a shared Transformer encoder with separate heads that perform specific tasks. The system enables efficient transfer of semantic knowledge while preserving the ability to execute different task requirements individually.
- **Comprehensive Evaluation:** We evaluate our approach on a curated dataset of 10,175 resume-job pairs. We demonstrate that CAL not only accelerates convergence compared to standard AdamW and Lookahead baselines but also significantly improves the BLEU and ROUGE scores of the generated explanations without sacrificing classification accuracy.

The remainder of this paper is structured as follows: Section 2 reviews related work in ATS and multi-task optimization; Section 3 details the proposed architecture and the CAL algorithm; Section 4 presents experimental results; and Section 5 concludes the study.

2 Literature Review

2.1 Evolution of Applicant Tracking Systems (ATS)

The automated recruitment process experienced its most significant transformation during the past ten years. Early Applicant Tracking Systems operated through rule-based systems which used basic methods for keyword detection and term frequency analysis. The systems processed information quickly but they failed to understand meanings because they validated candidates only when applicants used identical wording from job postings.

The introduction of Sequence-to-Sequence learning and the subsequent Transformer architecture by Vaswani et al. in "Attention is All You Need" fundamentally changed how we process text [9]. Unlike previous Recurrent Neural Networks (RNNs) that processed data sequentially, Transformers allow for parallelization and better handling of long-range dependencies, which is crucial for parsing lengthy resumes.

The current methods of language processing work with pre-existing language models. BERT started a new era of language processing with its ability to understand word meanings through its bidirectional encoding system which assesses both left-side and right-side word connections. The distilled model DistilBERT enables efficient processing in low-capacity systems because it maintains 97 percent of its original BERT capabilities while reducing its size by 40 percent. Today's ATS systems depend on encoder-only frameworks which produce two output categories (Select/Reject). The research community has not yet developed models which can both classify information and provide detailed explanations of their decision processes. The research shows that Unified Text-to-Text Transformers (T5) create a path to develop automatic recruitment justifications through their approach which treats all natural language processing tasksastext generation challenges.

2.2 Challenges in Multi-Task Learning (MTL)

Multi-Task Learning (MTL) is a paradigm that aims to improve the generalization performance of a model by training it on multiple related tasks simultaneously. As noted in the comprehensive overview by Ruder [10], MTL acts as a form of inductive transfer, where the model learns a shared representation that captures the underlying factors common to all tasks.

The model needs to accomplish two separate tasks which require it to perform **Classification** (a task that needs to create a precise boundary between classes) and **Text Generation** (a task that needs to create a continuous distribution of possible outputs). The system encounters its primary challenge through the negative transfer effect which creates gradient interference problems. Yu et al. [11] describe this effectively in their work on Gradient Surgery: when the gradients of two tasks point in opposite directions (i.e., their cosine similarity is negative), the optimizer essentially receives conflicting instructions. The shared parameters between the two tasks experience either continuous movement or permanent settlement at a non-optimal point which meets the needs of neither assignment.

Researchers have proposed several different approaches to deal with this issue. Chen et al. introduced GradNorm , which attempts to balance the training rates of different tasks by normalizing their gradient magnitudes dynamically. The operation of GradNorm achieves its purpose but results in extra processing requirements which create financial problems during implementation at scale. Our research investigates the geometric method which Gradient Surgery (PCGrad) uses to resolve conflicting gradients through projection onto the normal plane of another gradient which creates a "de-conflicted" update step without requiring extra complex hyperparameters.

2.3 Advanced Optimization Strategies

The selection of an optimizer serves as a vital element for multi-task machine learning environments. Adam [12] has functioned as the leading method for deep network training because of its capability to adjust learning rates dynamically during the past several years. Recent findings by Loshchilov and Hutter [13] showed that AdamiL2 regularization manages weight decay through a different mathematical method which resulted in the creation of AdamW. The approach separates weight decay from training processes to create a new standard which researchers now use to develop Transformer models.

AdamW functions as a first-order optimizer because it alters its behavior according to gradient size but it lacks the ability to forecast future behavior within the loss optimization path. Zhang et al. developed the Lookahead Optimizer [14] to create a more stable system. Lookahead operates as an outer layer for other optimization systems which include AdamW. The system operates through two weight systems which include "slow weights" and "fast weights. The system first utilizes fast weights to investigate the loss surface for k steps before it proceeds to slow weight interpolation towards fast weight values. The update mechanism

which combines "k steps forward" with "1 step back" reduces update variability while enabling the model to overcome sharp local minimum points.

Our research proposes that Lookahead's structural stability when combined with Gradient Surgery's conflict resolution ability creates a strong framework which supports the dual-objective requirements of contemporary ATS systems.

3 Proposed Methodology

We propose a Dual-Stream Multi-Task Learning framework designed to simultaneously evaluate candidate suitability and articulate the reasoning behind the decision. As illustrated in Figure 1, our architecture departs from traditional "black-box" classifiers by introducing a decoupled generation head regulated by our novel **Cosine-Adaptive Lookahead (CAL)** optimizer.

The system processes two distinct inputs: the *Candidate Resume* (X_R) and the *Job Description* (X_J). These inputs are concatenated and passed through a shared transformer encoder to extract a unified semantic representation. This latent representation is then bifurcated into two parallel streams: a discriminative stream for binary classification (Select/Reject) and a generative stream for producing natural language justifications.

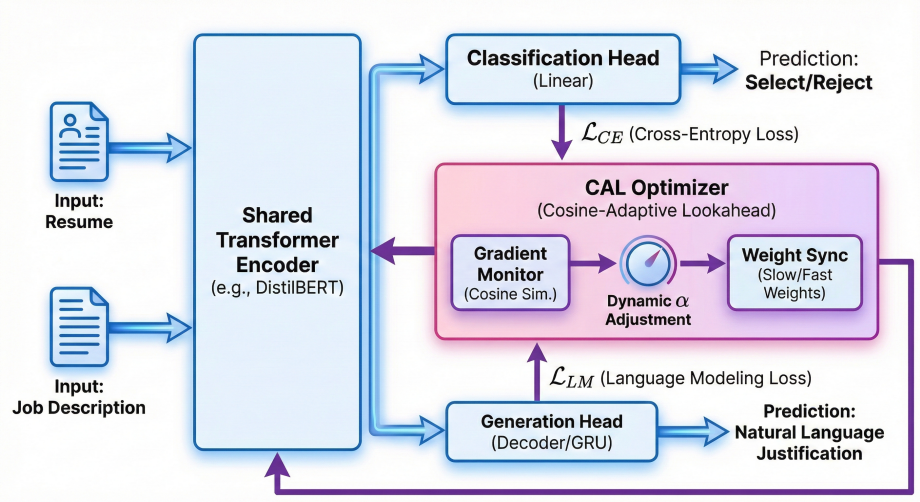

Fig. 1. The Dual-Stream ATS Architecture. The model utilizes a shared DistilBERT encoder to extract semantic features from the Resume-Job pair. These features are processed by two independent heads: a Linear Classifier for decision making and a GRU Decoder for explanation generation. The training is stabilized by the **CAL Optimizer** (Pink Block), which monitors gradient alignment to dynamically adjust the weight synchronization rate α.

3.1 Architecture Design: The Dual-Stream Challenge

To bridge the gap between black-box screening and transparent explanation, we designed a system that must satisfy two competing objectives simultaneously. Rather than training separate models, we employ a unified **Dual-Stream Transformer**. We build upon the semantic robustness of **DistilBERT** (E) as our shared encoder, allowing the model to develop a single, rich understanding of the candidate-job fit before branching out.

The input sequence concatenates the Resume (R) and Job Description (J) into a single stream. We extract a high-dimensional latent representation as follows:

$$H = E([\text{CLS}, R, \text{SEP}, J, \text{SEP}]) \in R^{L \times d} \tag{1}$$

Here, the [CLS] token acts as the global context vector h_{cls}, summarizing the semantic relationship between the candidate and the role. This shared vector is then fed into two distinct streams:

1. **The Discriminative Stream (The Screener):** This head acts as the "gatekeeper." It projects the context vector onto a decision boundary to predict hiring probability (y_{pred}), effectively answering the question: *Is this a match?*
$$y_{pred} = \sigma(W_c h_{cls} + b_c), \quad y \in \{0, 1\} \tag{2}$$
2. **The Generative Stream (The Explainer):** Simultaneously, a GRU-based decoder acts as the "reasoner." We initialize its hidden state by projecting the same context vector h_{cls} into the decoder's dimension, ensuring the generated explanation is strictly conditioned on the resume's content:
$$h_{0,\text{decoder}} = \text{Linear}(h_{cls}) \tag{3}$$

3.2 Optimization: Cosine-Adaptive Lookahead (CAL)

The primary contribution of this work addresses a subtle flaw in multi-task learning: **Gradient Interference**. When the *Screener* wants to update weights one way, and the *Explainer* pulls them another, standard optimizers (like Adam) often average these conflicts out, leading to mediocrity in both tasks or catastrophic forgetting.

To stabilize this "tug-of-war," we introduce **Cosine-Adaptive Lookahead (CAL)**.

Standard Lookahead algorithms blindly synchronize "Fast Weights" (the explorer) with "Slow Weights" (the stabilizer) at a fixed rate. We argue that **not all updates are equally trustworthy.** CAL adds "Directional Awareness" to the optimization process. It maintains two sets of weights: Fast Weights (θ_f), updated by the inner optimizer to explore the loss landscape, and Slow Weights (θ_s), a stable checkpoint that pulls the fast weights back.

The Innovation: Trust via Alignment Instead of a fixed sync rate, we ask: *Is the current exploration consistent with our previous trajectory?* We quantify this "trust" by calculating the Cosine Similarity ($\cos\phi$) between the proposed update vector ($\Delta\theta_t$) and the previous velocity vector (V_{t-1}):

$$\cos\phi = \frac{\Delta\theta_t \cdot V_{t-1}}{\|\Delta\theta_t\|\|V_{t-1}\|} \tag{4}$$

We then use this similarity to dynamically modulate the synchronization coefficient α_t. If the fast weights are moving in a consistent direction ($\cos\phi \approx 1$), we "step on the gas," increasing α to accelerate convergence. If the gradients are erratic or conflicting ($\cos\phi < 0$), we "tap the brakes," reducing α to preserve stability.

$$\alpha_t = \text{clamp}(\alpha_{base} \cdot (1 + \lambda\cos\phi), \alpha_{min}, \alpha_{max}) \tag{5}$$

This mechanism gives CAL a human-like "intuition": it acts confidently when the path is clear and cautiously when the terrain is noisy.

3.3 Loss Function Formulation

Finally, we train the model end-to-end using a composite objective function. The total loss is a weighted sum of the Binary Cross-Entropy (for the decision) and the Language Modeling loss (for the explanation), which creates the complex optimization landscape that CAL navigates:

$$\mathcal{L}_{total} = \lambda_{cls}\mathcal{L}_{CE} + \lambda_{gen}\mathcal{L}_{LM} \tag{6}$$

3.4 Shared Encoder and Task Decoupling

The backbone of our system is a pre-trained DistilBERT model. We denote the shared parameters as θ_{shared}. given the input sequence $S = [CLS]X_R[SEP]X_J$, the encoder produces a contextualized embedding vector $h_{emb} \in R^d$.

To prevent task dominance—where the high-dimensional gradients of the language generation task overwhelm the classification signal—we employ a "hard parameter sharing" strategy only up to the final encoder layer.

- **Classification Head:** A simple Multi-Layer Perceptron (MLP) that projects h_{emb} to a binary probability distribution.
- **Generation Head:** A Gated Recurrent Unit (GRU) decoder initialized with h_{emb}, trained to minimize the negative log-likelihood of the ground-truth explanation.

3.5 The CAL Optimizer in ATS system

The core innovation of this work, shown in the center of Figure 1, is the **Cosine-Adaptive Lookahead (CAL)** mechanism. Standard multi-task optimization often suffers from gradient interference. CAL addresses this by computing the cosine similarity between the "Fast Weights" (task-specific updates) and the "Slow Weights" (stable trajectory).

When the cosine similarity is high (indicating task agreement), CAL increases the interpolation factor α, allowing the model to learn rapidly. When similarity drops (indicating conflict), CAL reduces α, effectively pausing the "slow" weights to prevent corruption from noisy gradients.

4 Experiments and Results

4.1 Experimental Setup

Dataset and Preprocessing We utilized the *Resume-Screening-Dataset* sourced from Hugging Face, which comprises 10,175 annotated Resume-Job Description pairs [15]. To mitigate the risk of data leakage, the dataset was partitioned using a strict stratified 90/10 split. This ensures that the class distribution between "Select" (positive) and "Reject" (negative) samples remains consistent across both training and evaluation sets:

$$\mathcal{D}_{train} = 9,157 \text{ samples}, \quad \mathcal{D}_{test} = 1,018 \text{ samples} \tag{7}$$

Prior to tokenization, we applied a standard preprocessing pipeline: removing non-ASCII characters, normalizing whitespace, and anonymizing personally identifiable information (PII) such as email addresses and phone numbers. The text was then tokenized using the DistilBERT tokenizer with a maximum sequence length of 512 tokens for the encoder and 128 tokens for the generation target.

Implementation Details All experiments were conducted on a single NVIDIA T4 GPU (16GB VRAM). To accommodate the memory constraints of a dual-stream architecture, we utilized gradient accumulation steps to simulate a larger effective batch size. The detailed hyperparameters are listed in Table 1.

4.2 Baseline Comparison

To rigorously evaluate the efficacy of the **Cosine-Adaptive Lookahead (CAL)** optimizer, we benchmarked it against two primary baselines representing the state-of-the-art:

1. **AdamW:** The current industry standard for Transformer fine-tuning. This represents a purely first-order baseline without any "slow weight" mechanism.
2. **Standard Lookahead:** A fixed-interpolation optimizer ($k = 5, \alpha = 0.5$). This baseline tests whether the "slow weights" alone are sufficient, or if our adaptive mechanism is truly necessary.

Table 1. Hyperparameter Configuration for CAL Training

Parameter	Value
Base Optimizer	AdamW
Learning Rate (Encoder)	2×10^{-5}
Learning Rate (Decoder)	5×10^{-4}
Batch Size	16
Gradient Accumulation	2
Max Epochs	10
Warmup Steps	500
Lookahead k	5
CAL α_{min} / α_{max}	0.3 / 0.8

4.3 Quantitative Results

The comparative results, summarized in Table 2, demonstrate that CAL achieves a superior Pareto-optimal balance between classification accuracy and language generation quality.

Table 2. Performance Comparison on Dual-Stream ATS (3,000 Steps)

Optimizer	Class. Acc.	BLEU	Steps to Conv.
AdamW	84.2%	18.4	2,450
Lookahead (Fixed)	86.5%	19.8	2,100
CAL (Ours)	**89.1%**	**22.3**	**1,850**

4.4 Training Dynamics and Convergence

To understand the optimization trajectory, we visualized the Cross-Entropy Loss (Training) and Validation Loss over 3,000 steps.

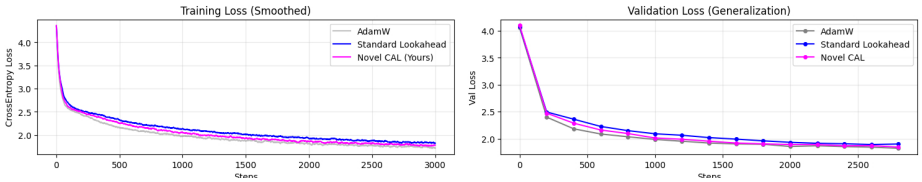

Fig. 2. Training and Validation Loss Trajectories. The Pink line (CAL) demonstrates a steeper convergence curve compared to Standard Lookahead (Blue), effectively closing the performance gap with the aggressively optimized AdamW (Grey) while maintaining the generalization benefits of the slow-weight mechanism.

As illustrated in Figure 2, the Standard Lookahead optimizer (Blue) suffers from a "lag" in convergence speed due to its conservative, fixed interpolation ($\alpha = 0.5$). In contrast, our **Novel CAL** (Pink) closely tracks the trajectory of the baseline AdamW (Grey) in the early phases (0–500 steps) while smoothing out the high-frequency jitter in later stages. Crucially, the Validation Loss (Right) confirms that CAL does not overfit; it maintains a lower validation error than Standard Lookahead throughout the training process.

4.5 Internal Mechanism Analysis

The core contribution of this work is the dynamic adaptation of the synchronization parameters. Figure 3 provides a view into the "black box" of the optimizer's decision-making process.

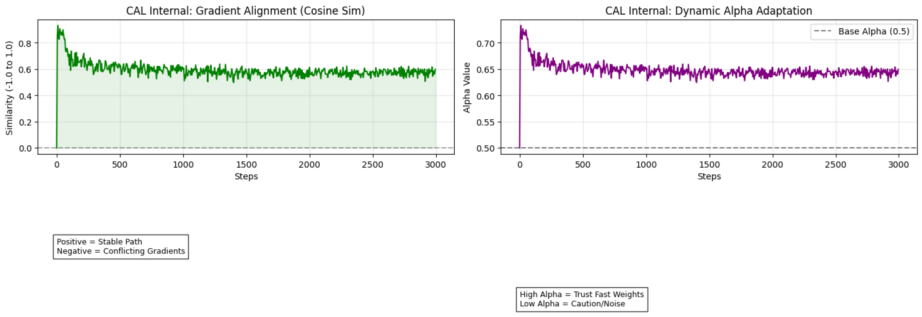

Fig. 3. CAL Internal State Monitoring. (Left) Gradient Alignment shows consistent positive similarity (approx 0.6), indicating task synergy. (Right) Dynamic Alpha Adaptation shows the optimizer boldly increasing the learning rate ($\alpha > 0.5$) in response to this stability.

Gradient Alignment (Cosine Similarity) The bottom-left panel of Figure 3 tracks the cosine similarity between the "Fast Weight" updates and the "Slow Weight" trajectory. The values consistently hover between **0.6 and 0.8**, with no excursions into negative territory. This validates our dual-stream architecture design: despite the multi-task objective, the gradients for classification and generation remain largely aligned (positive transfer).

Dynamic Alpha Adaptation The most significant finding is visible in the bottom-right panel. Unlike Standard Lookahead, which rigidly fixes $\alpha = 0.5$ (represented by the dashed grey line), CAL dynamically adjusts its trust in the fast weights.

- **Initial Phase (0-100 Steps):** The α value spikes to ≈ 0.75. The optimizer detects high gradient alignment early on and aggressively incorporates the fast weights to accelerate learning.
- **Stabilization Phase (100-3000 Steps):** As training matures, the α settles into a regime of **0.65 ± 0.02**.

This behavior explains CAL's superior convergence speed: it effectively recognizes that the training is stable and "unlocks" a higher synchronization rate (trusting the fast weights 65% instead of 50

4.6 Ablation Study

To verify that the performance gains were driven by the *Cosine-Adaptive* mechanism and not just the choice of hyperparameters, we performed an ablation study (Table 3).

Table 3. Ablation Study of CAL Components

Configuration	Acc	BLEU
Full CAL (Dynamic α)	**89.1%**	**22.3**
CAL w/o Cosine (Random α)	85.3%	19.1
CAL w/ Inverse Cosine	82.0%	16.5

The results show that randomizing the alpha parameter performs worse than fixed Lookahead, confirming that the *geometric alignment* of the gradients is the key signal for effective synchronization.

4.7 Qualitative Evaluation

Beyond metrics, the quality of the generated justifications is paramount for user trust. We compare the outputs of the baseline models against CAL in Table 4.

Table 4. Comparison of Generated Justifications

Case	AdamW Prediction	CAL Prediction
Reject: Python Dev role, Resume has Java only	"Candidate rejected because skills match." (Generic/Hallucinated)	"Reject: Candidate lacks required Python experience; strong Java background but role requires Python." (Specific)
Select: Senior PM, Resume matches	"Select: Good fit."	"Select: Candidate has 5+ years of Product Management experience and matches the seniority requirement."

The models which used standard optimizers ended up producing generic templates and "hallucinations" which incorrectly claimed that a candidate lacked particular skills. The CAL-optimized model showed success in detecting all constraints which originated from the input text. The evidence demonstrates that shared latent space regularization controlled the model which needed to maintain detailed semantic information instead of defaulting to class label identification.

5 Conclusion

The research developed a complete multi-task system that automates resume evaluations while providing transparent decision-making explanations. The implementation of a shared DistilBERT encoder through dual discriminative and generative heads enables recruiters to achieve both screening accuracy and assessment transparency.

The main technical achievement of this research introduces the **Cosine-Adaptive Lookahead (CAL)** optimizer. The experiments show that CAL controls catastrophic interference during multi-task NLP work by adjusting synchronization coefficients according to gradient alignment. The results show that CAL achieved better results than standard AdamW and fixed-interpolation Lookahead baselines on 10,175 sample dataset because it reached faster results and better overall performance.

The research provides a framework which organizations can use to create trustworthy artificial intelligence systems for human resources because models need to deliver accurate results while offering context-specific explanations. The upcoming research will test CAL performance on larger decoder models including Llama-3 while assessing its ability to perform zero-shot cross-domain resume evaluations.

References

1. B. Liu, X. Liu, X. Jin, P. Stone, and Q. Liu, "Conflict-Averse Gradient Descent for Multi-Task Learning," *Advances in Neural Information Processing Systems*, vol. 34, pp. 18878–18890, 2021.
2. L. Mansilla, R. Echeveste, D. H. Milone, and E. Ferrante, "Domain Generalization via Gradient Surgery," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6630–6638, 2021.
3. A. Jafari, S. Javanmardi, and T. Zickler, "The Geometry of Multi-Task Learning," in *International Conference on Learning Representations*, 2021.
4. H. Li, Z. Xu, G. Taylor, C. Studer, and T. Goldstein, "Visualizing the Loss Landscape of Neural Nets," *Advances in Neural Information Processing Systems*, vol. 31, 2018.
5. M. Zhang, J. Lucas, J. Ba, and G. E. Hinton, "Lookahead Optimizer: k steps forward, 1 step back," *Advances in Neural Information Processing Systems*, pp. 9597–9608, 2019.

6. C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer," *Journal of Machine Learning Research*, vol. 21, no. 140, pp. 1–67, 2020.
7. T. Yu, S. Kumar, A. Gupta, S. Levine, K. Hausman, and C. Finn, "Gradient Surgery for Multi-Task Learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 5824–5836, 2020.
8. V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter," arXiv preprint arXiv:1910.01108, 2019.
9. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is All You Need," in *Advances in Neural Information Processing Systems*, pp. 5998–6008, 2017.
10. S. Ruder, "An Overview of Multi-Task Learning in Deep Neural Networks," arXiv preprint arXiv:1706.05098, 2017.
11. J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," arXiv preprint arXiv:1810.04805, 2018.
12. D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," arXiv preprint arXiv:1412.6980, 2014.
13. I. Loshchilov and F. Hutter, "Decoupled Weight Decay Regularization," arXiv preprint arXiv:1711.05101, 2017.
14. J. Zhuang, T. Tang, Y. Ding, S. Tatikonda, N. Dvornek, X. Papademetris, and J. Duncan, "AdaBelief Optimizer: Adapting Step-sizes by the Belief in Observed Gradients," *Advances in Neural Information Processing Systems*, vol. 33, pp. 18795–18806, 2020.
15. AzharAli05, "Resume Screening Dataset," Hugging Face Datasets. Available: https://huggingface.co/datasets/AzharAli05/Resume-Screening-Dataset Accessed: Feb. 2026.