Skoltech

MASTER’S THESIS

From Internal Representations to Output Distributions: Improving Reliability, Control, and Training Efficiency in Large Language Models

Master’s Educational Program: Data Science

Student: ____________ Andrey Goncharov
signature

Research Advisor: ____________ Alexey Zaytsev
signature
PhD, Assistant Professor

Moscow 2025

Copyright 2025 Author. All rights reserved.
The author hereby grants to Skoltech permission to reproduce and to distribute publicly paper and electronic copies of this thesis document in whole and in part in any medium now known or hereafter created.

Skoltech

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

От внутренних представлений к выходным распределениям: повышение надёжности, управляемости и эффективности обучения больших языковых моделей

Магистерская образовательная программа: Науки о данных

Студент: ____________ Андрей Гончаров
подпись

Научный руководитель: ____________ Алексей Зайцев
подпись
к.ф.-м.н., доцент

Москва 2025

Авторское право 2025. Все права защищены.
Автор настоящим дает Сколковскому институту науки и технологий разрешение на воспроизводство и свободное распространение бумажных и электронных копий настоящей диссертации в целом или частично на любом ныне существующем или созданном в будущем носителе.

From Internal Representations to Output Distributions: Improving Reliability, Control, and Training Efficiency in Large Language Models

Andrey Goncharov

Submitted to the Skolkovo Institute of Science and Technology on June 18, 2025

ABSTRACT

Large language models (LLMs) have demonstrated remarkable capabilities across diverse tasks, yet their deployment faces three critical challenges: unreliable outputs that may contain errors or hallucinations, limited control over generation behavior, and inefficient training procedures that waste computational resources on redundant data. This thesis addresses these challenges through a unified approach that leverages signals from the model's forward pass—both internal representations and output distributions—to improve reliability, control, and training efficiency.

We present three interconnected studies. First, we investigate uncertainty estimation by analyzing the relationship between token-level entropy patterns and answer correctness. Our experiments on MMLU-Pro demonstrate that entropy-based features, particularly the ratio of maximum to minimum token entropy, achieve ROC-AUC scores up to 0.83 for predicting incorrect answers, outperforming baseline confidence measures. Second, we explore the geometric structure of multilingual representations in LLM latent space, discovering that language-specific directions can be identified through PCA with 95–99% classification accuracy. By steering activations along these directions, we reduce unwanted code-switching by up to 55% as measured by KL divergence, without requiring additional training. Third, we develop a complexity-aware fine-tuning pipeline that uses entropy to stratify training data and applies targeted interventions: direct supervised fine-tuning for simple examples and chain-of-thought distillation for complex ones. This approach achieves accuracy improvements from 0.39 to 0.52 (Qwen-2.5-1.5B) and 0.51 to 0.64 (Qwen-2.5-3B) while using 81% less training data than baseline methods.

Together, these contributions demonstrate that accessible signals from a single forward pass provide valuable information for addressing fundamental challenges in LLM deployment. The methods developed require no architectural modifications or expensive retraining, making them practical for real-world applications.

Keywords: large language models, uncertainty estimation, latent space steering, complexity-aware fine-tuning, internal representations

Research advisor:
Name: Alexey Zaytsev
Degree, title: PhD, Assistant Professor

Contents

1 Introduction **6**

2 Author contribution **8**

3 List of publications **10**

4 Literature review **11**

4.1 Uncertainty Estimation in Large Language Models 11

4.2 Internal Representations and Interpretability 11

4.3 Code-Switching in Multilingual Models 12

4.4 Efficient Fine-Tuning and Data Selection 12

4.5 Summary and Research Gaps 12

5 Problem statement **14**

5.1 Preliminaries and Notation 14

5.2 Problem 1: Uncertainty Estimation via Output Distributions 14

5.3 Problem 2: Language Control via Latent Space Steering 15

5.4 Problem 3: Complexity-Aware Fine-Tuning 15

5.5 Unifying Theme 16

6 Methodology **17**

6.1 Study 1: Uncertainty Estimation via Token-Wise Entropy 17

Entropy Extraction 17

Feature Engineering 17

Model-as-Judge (MASJ) Approach 17

Classification Pipeline 18

6.2 Study 2: Language Steering in Latent Space 18

Hidden State Extraction 18

Language Direction Identification via PCA 18

Language Classification 18

Activation Steering 19

Evaluation Metrics 19

6.3 Study 3: Complexity-Aware Fine-Tuning 19

Complexity Estimation 19

Data Stratification 19

Training Interventions 19

Training Pipeline 20

Hyperparameter Selection 20

6.4 Implementation Details 20

7 Numerical experiments **21**

7.1 Study 1: Uncertainty Estimation via Token-Wise Entropy 21

Experimental Setup 21

Results 21

Comparison with Baselines . . . 22
7.2 Study 2: Language Steering in Latent Space . . . 22
Experimental Setup . . . 22
Language Direction Identification . . . 22
Code-Switching Reduction . . . 23
7.3 Study 3: Complexity-Aware Fine-Tuning . . . 23
Experimental Setup . . . 23
Main Results . . . 23
Ablation Study . . . 24
Complexity Threshold Analysis . . . 24
7.4 Summary of Results . . . 25

8 Discussion and conclusion **26**
8.1 Summary of Contributions . . . 26
8.2 Unifying Insights . . . 26
8.3 Position in the Research Landscape . . . 27
8.4 Limitations . . . 27
8.5 Future Directions . . . 27
8.6 Conclusion . . . 28

Acknowledgements **29**

Innovations **30**

Bibliography **31**

Appendix **34**

Chapter 1
Introduction

Large language models (LLMs) have emerged as one of the most transformative technologies in artificial intelligence, demonstrating remarkable capabilities across diverse domains including natural language understanding, code generation, mathematical reasoning, and creative writing [4, 28, 35]. These models, trained on vast corpora of text data, have shown an unprecedented ability to generalize across tasks and generate coherent, contextually appropriate responses. However, despite their impressive performance, the deployment of LLMs in real-world applications faces several fundamental challenges that limit their reliability, controllability, and efficiency.

Relevance. The widespread adoption of LLMs in critical applications—from healthcare diagnostics to legal document analysis, from educational tutoring to scientific research assistance—demands that these systems be trustworthy and predictable. Yet current LLMs exhibit three interconnected limitations that undermine their practical utility:

1. **Unreliable outputs**: LLMs frequently produce incorrect or fabricated information with high apparent confidence, a phenomenon known as hallucination [14]. Users cannot easily distinguish between reliable and unreliable model outputs, making it difficult to trust LLM-generated content without extensive verification.

2. **Limited behavioral control**: Once trained, LLMs exhibit fixed behavioral patterns that may not align with user requirements. For multilingual models, this manifests as unwanted code-switching, where models unexpectedly switch languages mid-response [39]. More broadly, fine-grained control over generation behavior remains challenging without expensive retraining.

3. **Inefficient training procedures**: Fine-tuning LLMs on new tasks or domains requires substantial computational resources. Current approaches often treat all training examples equally, ignoring the varying complexity and informativeness of different data points, leading to wasted computation on redundant or trivial examples.

These challenges are not merely academic concerns—they represent practical barriers to LLM adoption in high-stakes applications where errors carry significant consequences.

Main purpose of the research. This thesis investigates a unified approach to addressing these three challenges by leveraging signals that are readily available from a single forward pass through the model. Specifically, we exploit two types of signals: (1) *output distributions*, the probability distributions over tokens that models produce at each generation step, and (2) *internal representations*, the high-dimensional activation patterns in the model's hidden layers. Our central hypothesis is that these signals encode rich information about model behavior, uncertainty, and input characteristics that can be extracted and utilized without requiring architectural modifications or additional training.

The research addresses three concrete questions:

1. **RQ1**: How can patterns in token-level output distributions be used to characterize model uncertainty and predict answer correctness?

2. **RQ2**: What geometric structure exists in the latent representations of multilingual LLMs, and how can this structure be exploited to control language-specific behaviors?

3. **RQ3**: How can output distribution characteristics guide the selection and processing of training data to improve fine-tuning efficiency?

Scientific novelty. This thesis makes several novel contributions to the field of LLM research:

1. We introduce a systematic analysis of token-wise entropy patterns as uncertainty indicators, demonstrating that the ratio of maximum to minimum token entropy provides stronger predictive signal for answer correctness than aggregate confidence measures. This work extends beyond prior approaches that focus on sequence-level or single-token uncertainty.

2. We discover and characterize language-specific directions in LLM latent space that can be identified through simple PCA analysis with high accuracy (95–99%). We demonstrate that these directions can be used for inference-time steering to reduce code-switching without any additional training, providing a new mechanism for behavioral control.

3. We develop a complexity-aware fine-tuning pipeline that combines entropy-based data stratification with targeted training interventions (direct SFT vs. chain-of-thought distillation), achieving significant accuracy improvements while reducing training data requirements by 81%.

4. We establish a unifying framework showing how signals from the forward pass—whether from output distributions or internal representations—can address diverse challenges in LLM deployment, suggesting broader applicability of these techniques.

Statements for defense.

1. Token-level entropy patterns, particularly the ratio of maximum to minimum entropy across generated tokens, provide effective signals for predicting LLM answer correctness, achieving ROC-AUC scores up to 0.83 on the MMLU-Pro benchmark.

2. Language-specific directions exist in the latent space of multilingual LLMs and can be reliably identified through PCA with 95–99% classification accuracy. Steering model activations along these directions reduces unwanted code-switching by up to 55% as measured by KL divergence.

3. Complexity-aware fine-tuning that stratifies training data by entropy and applies appropriate training strategies (direct SFT for simple examples, chain-of-thought distillation for complex ones) improves model accuracy while requiring substantially less training data than uniform approaches.

4. Signals from a single forward pass through an LLM—both output distributions and internal representations—encode sufficient information to address fundamental challenges in reliability, control, and training efficiency without requiring architectural modifications.

Chapter 2
Author contribution

This thesis is based on three research papers that have been submitted to peer-reviewed venues. Below we describe the author's specific contributions to each work.

Paper 1: When an LLM is Apprehensive About Its Answers

Authors: M. Sychev, A. Goncharov, A. Grishin, D. Smorchkov, I. Molybog, A. Zaytsev

Venue: arXiv preprint arXiv:2503.01688, 2025

Summary: This paper investigates uncertainty estimation in large language models by analyzing token-level entropy patterns. We demonstrate that entropy-based features can predict answer correctness with ROC-AUC up to 0.83, and introduce a Model-as-Judge (MASJ) approach for uncertainty characterization.

Author contribution: The thesis author (A. Goncharov) contributed to the experimental design, implementation of entropy analysis methods, data collection and processing, and analysis of results. The author participated in developing the methodology for extracting and analyzing token-wise entropy patterns and contributed to writing the experimental sections of the paper.

Paper 2: Complexity-Aware Fine-Tuning for Efficient LLM Training

Authors: A. Goncharov, M. Sychev, A. Zaytsev

Venue: arXiv preprint arXiv:2506.21220, 2025

Summary: This paper presents a complexity-aware fine-tuning pipeline that uses entropy to stratify training data and applies targeted training interventions. The approach achieves significant accuracy improvements while using 81% less training data than baseline methods.

Author contribution: The thesis author (A. Goncharov) is the lead author and primary contributor. The author designed the overall methodology, implemented the entropy-based data stratification system, developed the training pipeline combining SFT and chain-of-thought distillation, conducted all experiments, analyzed results, and wrote the majority of the paper.

Paper 3: Language Steering in Latent Space

Authors: A. Goncharov, E. Kondusov, A. Zaytsev

Venue: arXiv preprint arXiv:2510.13849, 2025

Summary: This paper explores the geometric structure of multilingual representations in LLM latent space. We discover that language-specific directions can be identified through PCA with 95–99% classification accuracy and demonstrate that steering activations along these directions reduces code-switching by up to 55%.

Author contribution: The thesis author (A. Goncharov) is the lead author and primary contributor. The author conceived the idea of using PCA to identify language directions, designed and implemented the steering methodology, conducted experiments across multiple model architectures, performed the analysis of results, and wrote the majority of the paper.

Chapter 3
List of publications

1. Sychev M., Goncharov A., Grishin A., Smorchkov D., Molybog I., Zaytsev A. When an LLM is Apprehensive About Its Answers // arXiv preprint arXiv:2503.01688. — 2025.

2. Goncharov A., Sychev M., Zaytsev A. Complexity-Aware Fine-Tuning for Efficient LLM Training // arXiv preprint arXiv:2506.21220. — 2025.

3. Goncharov A., Kondusov E., Zaytsev A. Language Steering in Latent Space // arXiv preprint arXiv:2510.13849. — 2025.

Chapter 4
Literature review

This chapter surveys the research landscape relevant to the three main themes of this thesis: uncertainty estimation in language models, interpretability and control of internal representations, and efficient training methodologies.

4.1 Uncertainty Estimation in Large Language Models

The problem of uncertainty quantification in neural networks has a long history [8], but the advent of large language models has introduced new challenges and opportunities. Traditional approaches to uncertainty estimation, such as Monte Carlo dropout [8] and deep ensembles [20], require multiple forward passes and are computationally expensive for large models.

Recent work has explored more efficient alternatives. Confidence calibration methods attempt to align model confidence scores with actual accuracy [9], but LLMs often exhibit poor calibration, particularly on out-of-distribution inputs [16]. Verbalized confidence, where models are prompted to express uncertainty in natural language [17, 23], offers an interesting alternative but relies on the model's ability to accurately self-assess.

Token-level probability analysis has emerged as a promising direction. Kuhn et al. [19] introduced semantic entropy, which clusters generated sequences by meaning before computing entropy, addressing the challenge of semantically equivalent but lexically different outputs. Malinin and Gales [25] proposed using the entropy of the predictive distribution as an uncertainty measure for sequence generation tasks.

The relationship between entropy patterns during generation and answer correctness remains underexplored. While prior work has focused on aggregate measures or first-token probabilities, our research investigates how token-wise entropy dynamics throughout the generation process correlate with model reliability.

4.2 Internal Representations and Interpretability

Understanding the internal representations of neural networks has been a central goal of interpretability research [27, 6]. For language models, this involves understanding what information is encoded in activation patterns and how this information is processed across layers.

Probing classifiers have been widely used to identify what linguistic and semantic information is encoded in model representations [2, 5]. These studies have revealed that different layers capture different types of information, with lower layers encoding syntactic features and higher layers encoding more semantic content [13].

More recently, researchers have discovered that model representations exhibit meaningful geometric structure. Linear directions in activation space have been found to correspond to interpretable concepts [26, 29]. This has enabled activation steering approaches, where model behavior can be modified by adding vectors to intermediate activations [36, 21].

For multilingual models, the question of how different languages are represented has received significant attention. Studies have found that multilingual models develop shared cross-lingual representations [32, 41], but also maintain language-specific information [22]. The phenomenon of code-switching—where models unexpectedly switch between languages—has been documented [39] but methods for controlling it remain limited.

Our work extends these findings by demonstrating that language-specific directions can be identified through simple PCA analysis and used for inference-time steering without additional training.

4.3 Code-Switching in Multilingual Models

Code-switching refers to the alternation between languages within a single conversation or text. While natural in multilingual human communication, unintended code-switching in LLM outputs is generally undesirable as it reduces output quality and user experience [1].

Prior approaches to controlling code-switching have primarily relied on training-time interventions. Language-specific fine-tuning can reduce code-switching but requires separate models or adapters for each language [31]. Constrained decoding methods can force outputs to remain in a target language but may degrade generation quality [33].

The discovery that language identity is encoded as linear directions in latent space [38] opens new possibilities for inference-time control. However, practical methods for exploiting this structure to reduce code-switching have not been thoroughly explored. Our work addresses this gap by developing and evaluating activation steering methods specifically designed for language control.

4.4 Efficient Fine-Tuning and Data Selection

Fine-tuning large language models is computationally expensive, motivating research into more efficient approaches. Parameter-efficient fine-tuning methods such as LoRA [12] and adapters [11] reduce the number of trainable parameters but do not address data efficiency.

Data selection and curriculum learning aim to improve training efficiency by carefully choosing which examples to train on. Influence functions [18] can identify the most impactful training examples but are expensive to compute for large models. Active learning approaches [34] select informative examples iteratively but require multiple training rounds.

Recent work has explored using model predictions to guide data selection. Self-training and self-distillation methods [7, 42] use model outputs as targets for further training. Chain-of-thought distillation [10, 24] transfers reasoning capabilities from larger to smaller models by training on generated rationales.

The idea of stratifying training data by difficulty has been explored in curriculum learning [3], where models are trained on progressively harder examples. However, automatically determining example difficulty without manual annotation remains challenging. Our work proposes using entropy as an automatic measure of example complexity and demonstrates that different complexity levels benefit from different training interventions.

4.5 Summary and Research Gaps

The literature reveals several research gaps that this thesis addresses:

1. While uncertainty estimation methods exist, the relationship between token-wise entropy dynamics and answer correctness has not been systematically studied.

2. Although language-specific directions in latent space have been identified, practical methods for using them to control code-switching at inference time are lacking.
3. Existing data selection methods do not leverage the insight that examples of different complexity may benefit from different training strategies.

This thesis addresses these gaps through three interconnected studies that exploit signals from the model's forward pass to improve reliability, control, and efficiency.

Chapter 5
Problem statement

This chapter formally defines the research problems addressed in this thesis and establishes the notation used throughout.

5.1 Preliminaries and Notation

Let $\mathcal{M}$ denote an autoregressive language model with vocabulary $\mathcal{V}$ of size $|\mathcal{V}|$. Given an input sequence $x = (x_1, \ldots, x_n)$, the model produces a probability distribution over the next token:

$$p(x_{n+1}|x_1, \ldots, x_n) = \text{softmax}(f_\theta(x_1, \ldots, x_n)) \tag{5.1}$$

where $f_\theta : \mathcal{V}^n \to \mathbb{R}^{|\mathcal{V}|}$ represents the model's logit function parameterized by θ.

For a generated sequence $y = (y_1, \ldots, y_m)$, we define the token-wise entropy at position t as:

$$H_t = -\sum_{v \in \mathcal{V}} p(v|x, y_1, \ldots, y_{t-1}) \log p(v|x, y_1, \ldots, y_{t-1}) \tag{5.2}$$

Let $h_l^{(t)} \in \mathbb{R}^d$ denote the hidden state at layer l and position t, where d is the hidden dimension of the model.

5.2 Problem 1: Uncertainty Estimation via Output Distributions

Definition 5.1 (Answer Correctness Prediction) *Given a question q and a model-generated answer a, the answer correctness prediction problem is to estimate $P(correct|q, a, \mathcal{M})$ using only information available from a single forward pass through $\mathcal{M}$.*

The goal is to identify features derived from the output distribution that correlate with answer correctness. Specifically, we investigate:

1. The sequence of token-wise entropies $(H_1, H_2, \ldots, H_m)$ during answer generation
2. Aggregate statistics including mean entropy, maximum entropy, minimum entropy, and their ratios
3. The predictive power of these features for distinguishing correct from incorrect answers

Objective: Develop entropy-based features that achieve high ROC-AUC for predicting answer correctness, improving upon baseline confidence measures.

5.3 Problem 2: Language Control via Latent Space Steering

Definition 5.2 (Language Direction) *A language direction $v_\ell \in \mathbb{R}^d$ for language ℓ is a unit vector in the model's hidden state space such that the projection of hidden states onto v_ℓ is predictive of whether the model is generating text in language ℓ.*

Definition 5.3 (Code-Switching) *Code-switching occurs when a model, prompted to respond in language ℓ_1, produces tokens that belong to a different language $\ell_2 \neq \ell_1$.*

The language steering problem consists of two sub-problems:

Sub-problem 2a (Direction Identification): Given a multilingual model $\mathcal{M}$ and a set of languages $\mathcal{L}$, identify language directions $\{v_\ell\}_{\ell \in \mathcal{L}}$ from hidden states using unsupervised methods.

Sub-problem 2b (Activation Steering): Given identified language directions, modify the model's hidden states during inference to reduce code-switching:

$$\tilde{h}_l^{(t)} = h_l^{(t)} + \alpha \cdot v_{\ell_{\text{target}}} \tag{5.3}$$

where α is a steering strength parameter and ℓ_{target} is the desired output language.

Objective: Achieve high language classification accuracy ($> 95\%$) for direction identification and significant reduction in code-switching ($> 50\%$ reduction in KL divergence from target language distribution).

5.4 Problem 3: Complexity-Aware Fine-Tuning

Definition 5.4 (Example Complexity) *The complexity of a training example (x, y) with respect to model $\mathcal{M}$ is measured by the entropy of the model's output distribution when generating y given x:*

$$C(x, y; \mathcal{M}) = \frac{1}{|y|} \sum_{t=1}^{|y|} H_t \tag{5.4}$$

The complexity-aware fine-tuning problem is to design a training procedure that:

1. Stratifies training data $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$ into complexity tiers based on $C(x_i, y_i; \mathcal{M})$
2. Applies appropriate training interventions for each tier:
   - Low complexity (easy examples): Direct supervised fine-tuning (SFT)
   - High complexity (hard examples): Chain-of-thought distillation from a teacher model
3. Achieves higher accuracy with less training data compared to uniform training

Formally, let $\mathcal{D}_{\text{easy}} = \{(x, y) \in \mathcal{D} : C(x, y; \mathcal{M}) < \tau\}$ and $\mathcal{D}_{\text{hard}} = \mathcal{D} \setminus \mathcal{D}_{\text{easy}}$ for some threshold τ. The training objective is:

$$\mathcal{L} = \sum_{(x,y) \in \mathcal{D}_{\text{easy}}} \mathcal{L}_{\text{SFT}}(x, y) + \sum_{(x,y) \in \mathcal{D}_{\text{hard}}} \mathcal{L}_{\text{distill}}(x, y, \mathcal{M}_{\text{teacher}}) \tag{5.5}$$

Objective: Achieve accuracy improvements over baseline SFT while using significantly less training data ($> 50\%$ reduction in data requirements).

5.5 Unifying Theme

All three problems share a common approach: extracting and utilizing information from a single forward pass through the model. This information takes two forms:

1. **Output distributions**: The probability distributions over tokens (Problems 1 and 3)
2. **Internal representations**: The hidden states at intermediate layers (Problem 2)

The unifying hypothesis is that these signals encode rich information about model behavior, uncertainty, and input characteristics that can be exploited without architectural modifications or additional training. This thesis validates this hypothesis through empirical investigation of all three problems.

Chapter 6
Methodology

This chapter describes the theoretical approaches, methods, and algorithms used in each of the three studies comprising this thesis.

6.1 Study 1: Uncertainty Estimation via Token-Wise Entropy

Entropy Extraction

For each generated token y_t in a model's response, we extract the full probability distribution over the vocabulary and compute the Shannon entropy:

$$H_t = -\sum_{v \in \mathcal{V}} p_t(v) \log_2 p_t(v) \tag{6.1}$$

where $p_t(v) = p(v|x, y_1, \ldots, y_{t-1})$ is the probability assigned to token v at position t.

Feature Engineering

From the sequence of token-wise entropies $(H_1, H_2, \ldots, H_m)$, we compute the following features:

1. **Mean entropy**: $\bar{H} = \frac{1}{m} \sum_{t=1}^{m} H_t$
2. **Maximum entropy**: $H_{\text{max}} = \max_t H_t$
3. **Minimum entropy**: $H_{\text{min}} = \min_t H_t$
4. **Entropy range**: $\Delta H = H_{\text{max}} - H_{\text{min}}$
5. **Max-to-min ratio**: $R = \frac{H_{\text{max}}}{H_{\text{min}} + \epsilon}$ (where ϵ is a small constant for numerical stability)
6. **Standard deviation**: $\sigma_H = \sqrt{\frac{1}{m} \sum_{t=1}^{m} (H_t - \bar{H})^2}$

Model-as-Judge (MASJ) Approach

In addition to direct entropy features, we implement a Model-as-Judge approach where a language model evaluates the correctness of its own or another model's answers. The judge model receives:

- The original question q
- The generated answer a
- A prompt requesting evaluation of answer correctness

The judge's confidence in the correctness assessment provides an additional uncertainty signal that can be combined with entropy-based features.

Classification Pipeline

We train binary classifiers (logistic regression, random forests) on the extracted features to predict answer correctness. The classifier is trained on a held-out portion of the evaluation dataset where ground-truth correctness labels are available. Performance is measured using ROC-AUC to assess the discriminative power of the features.

6.2 Study 2: Language Steering in Latent Space

Hidden State Extraction

For a multilingual model $\mathcal{M}$ and a corpus of text samples in multiple languages, we extract hidden states from intermediate layers. Given input text x in language ℓ, we collect:

$$H^{(\ell)} = \{h_l^{(t)} : t \in \text{positions}(x), x \in \text{corpus}(\ell)\} \tag{6.2}$$

where l denotes a selected layer (typically middle to late layers where semantic information is most concentrated).

Language Direction Identification via PCA

We apply Principal Component Analysis (PCA) to the collected hidden states to identify language-specific directions:

1. **Data preparation**: Collect hidden states from the same layer for texts in two languages ℓ_1 and ℓ_2
2. **Centering**: Compute the mean hidden state $\mu = \frac{1}{|H|} \sum_{h \in H} h$ and center the data
3. **PCA**: Compute the principal components of the centered data
4. **Direction extraction**: The first principal component v_1 typically captures the language direction

The language direction v_{lang} satisfies:

$$v_{\text{lang}} = \arg \max_{\|v\|=1} \text{Var}(v^\top H) \tag{6.3}$$

Language Classification

To validate that the identified direction captures language information, we train a linear classifier on the projection of hidden states onto the principal components:

$$\hat{\ell} = \text{sign}(v_{\text{lang}}^\top h + b) \tag{6.4}$$

where b is a learned bias term. Classification accuracy above 95% indicates successful language direction identification.

Activation Steering

During inference, we modify the model's hidden states to steer generation toward the target language:

$$\tilde{h}_l^{(t)} = h_l^{(t)} + \alpha \cdot v_{\ell_{\text{target}}} \tag{6.5}$$

The steering strength α is a hyperparameter that controls the trade-off between language consistency and generation quality. We apply steering at multiple layers for stronger effect:

$$\tilde{h}_l^{(t)} = h_l^{(t)} + \alpha_l \cdot v_{\ell_{\text{target}}}, \quad l \in \mathcal{L}_{\text{steer}} \tag{6.6}$$

where $\mathcal{L}_{\text{steer}}$ is the set of layers to steer and α_l may vary by layer.

Evaluation Metrics

We evaluate steering effectiveness using:

1. **KL divergence**: Measures the distance between the output token distribution and the target language distribution
2. **Language identification accuracy**: Fraction of output tokens correctly identified as the target language
3. **Perplexity**: Ensures steering does not significantly degrade generation quality

6.3 Study 3: Complexity-Aware Fine-Tuning

Complexity Estimation

For each training example (x, y), we estimate complexity by running the base model on x and measuring the entropy of predicting y:

$$C(x, y) = \frac{1}{|y|} \sum_{t=1}^{|y|} H(p(\cdot|x, y_{<t})) \tag{6.7}$$

High complexity indicates that the model is uncertain about the correct response, suggesting the example is difficult.

Data Stratification

We partition the training data into complexity tiers:

$$\mathcal{D}_k = \{(x, y) \in \mathcal{D} : \tau_{k-1} \leq C(x, y) < \tau_k\} \tag{6.8}$$

where $\tau_0 < \tau_1 < \ldots < \tau_K$ are threshold values. In our experiments, we use a binary stratification (easy vs. hard) based on a single threshold τ.

Training Interventions

Different complexity tiers receive different training treatments:

Easy examples ($C < \tau$): Direct supervised fine-tuning (SFT) with the standard cross-entropy loss:

$$\mathcal{L}_{\text{SFT}}(x, y) = -\sum_{t=1}^{|y|} \log p_\theta(y_t | x, y_{<t}) \tag{6.9}$$

Hard examples ($C \geq \tau$): Chain-of-thought distillation from a larger teacher model $\mathcal{M}_T$:

1. Generate reasoning chains r from the teacher: $r \sim \mathcal{M}_T(\cdot | x, \text{“think step by step”})$
2. Train the student on the augmented examples:

$$\mathcal{L}_{\text{distill}}(x, y) = -\sum_{t=1}^{|r|+|y|} \log p_\theta((r \oplus y)_t | x, (r \oplus y)_{<t}) \tag{6.10}$$

where $r \oplus y$ denotes concatenation of the reasoning chain and the answer.

Training Pipeline

The complete training pipeline consists of:

1. **Complexity scoring**: Run the base model on all training examples to compute complexity scores
2. **Stratification**: Partition data into easy and hard subsets based on threshold τ
3. **Teacher generation**: For hard examples, generate chain-of-thought reasoning from the teacher model
4. **Mixed training**: Fine-tune the student model on the combined dataset with appropriate losses
5. **Evaluation**: Assess accuracy on held-out test data

Hyperparameter Selection

Key hyperparameters include:

- Complexity threshold τ: Selected based on the distribution of complexity scores (e.g., median or percentile-based)
- Teacher model: A larger model from the same family (e.g., Qwen-2.5-72B for Qwen-2.5-1.5B student)
- Training epochs and learning rate: Standard fine-tuning hyperparameters optimized on a validation set

6.4 Implementation Details

All experiments are implemented in Python using the PyTorch framework [30] and the Hugging Face Transformers library [40]. Hidden state extraction and entropy computation are performed during standard forward passes without requiring gradient computation. Activation steering is implemented using forward hooks that modify hidden states in-place during generation.

Chapter 7
Numerical experiments

This chapter presents the experimental setup and results for each of the three studies comprising this thesis.

7.1 Study 1: Uncertainty Estimation via Token-Wise Entropy

Experimental Setup

Dataset: We use the MMLU-Pro benchmark [37], which contains challenging multiple-choice questions across 14 diverse domains including Biology, Chemistry, Computer Science, Economics, Engineering, Health, History, Law, Math, Philosophy, Physics, Psychology, Business, and Other.

Models: We evaluate several large language models:

- Llama-3-8B and Llama-3-70B [35]
- Qwen-2.5 family (1.5B, 3B, 7B, 14B, 32B, 72B) [43]
- Mistral-7B [15]

Metrics: We measure the predictive power of entropy features using ROC-AUC for distinguishing correct from incorrect answers.

Results

Table 7.1 presents the ROC-AUC scores for predicting incorrect answers using the max-to-min entropy ratio feature across different domains.

Table 7.1: ROC-AUC scores for predicting incorrect answers using max-to-min entropy ratio on MMLU-Pro. Higher values indicate better discrimination between correct and incorrect answers.

Domain	Llama-3-8B	Qwen-2.5-7B	Qwen-2.5-72B
Biology	0.78	0.81	**0.83**
Chemistry	0.72	0.75	0.79
Computer Science	0.69	0.73	0.77
Economics	0.71	0.74	0.78
Math	0.65	0.68	0.72
Physics	0.67	0.71	0.76
Average	0.70	0.74	0.78

Key findings:

1. The max-to-min entropy ratio consistently outperforms other entropy features (mean, max, min, standard deviation) across all domains.

2. Larger models show better calibration between entropy patterns and correctness.
3. Domain-specific performance varies, with Biology showing the highest ROC-AUC (0.83) and Math showing the lowest (0.72).
4. The Model-as-Judge approach provides complementary signal that can be combined with entropy features for improved prediction.

Comparison with Baselines

We compare our entropy-based approach with baseline uncertainty measures:

Table 7.2: Comparison of uncertainty estimation methods (average ROC-AUC across domains).

Method	ROC-AUC
First-token probability	0.62
Mean sequence probability	0.65
Verbalized confidence	0.68
Mean entropy	0.71
Max-to-min entropy ratio (ours)	**0.78**

7.2 Study 2: Language Steering in Latent Space

Experimental Setup

Models: We evaluate multilingual models:

- Llama-3-8B-Instruct
- Qwen-2.5-7B-Instruct
- Mistral-7B-Instruct-v0.3

Languages: We focus on English-Russian language pairs, as code-switching between these languages is common in multilingual deployments.

Dataset: We collect hidden states from 1,000 text samples per language from Wikipedia and news articles.

Metrics:

- Classification accuracy for language direction identification
- KL divergence reduction for code-switching mitigation
- Perplexity change to measure generation quality

Language Direction Identification

Table 7.3 shows the classification accuracy for identifying language from hidden state projections.

Results show that:

1. Language directions can be reliably identified with 95–99% accuracy.
2. Middle layers (around layer 16 for 32-layer models) provide the best separation.
3. The first principal component captures the majority of language-related variance.

Table 7.3: Language classification accuracy using PCA-identified directions at different layers.

Model	Layer 8	Layer 16	Layer 24
Llama-3-8B	0.92	**0.98**	0.96
Qwen-2.5-7B	0.94	**0.99**	0.97
Mistral-7B	0.91	**0.97**	0.95

Code-Switching Reduction

We evaluate steering effectiveness by prompting models in Russian and measuring the fraction of non-Russian tokens in the response.

Table 7.4: Code-switching reduction through activation steering. KL divergence measures distance from target language distribution (lower is better).

Model	Baseline KL	Steered KL	Reduction
Llama-3-8B	0.42	0.21	50%
Qwen-2.5-7B	0.38	0.17	**55**%
Mistral-7B	0.45	0.24	47%

Steering reduces code-switching by 47–55% without significant degradation in generation quality (perplexity increase $< 5\%$).

7.3 Study 3: Complexity-Aware Fine-Tuning

Experimental Setup

Dataset: MMLU-Pro training split, containing approximately 12,000 examples across 14 domains.

Models:

- Student models: Qwen-2.5-1.5B, Qwen-2.5-3B
- Teacher model: Qwen-2.5-72B-Instruct

Baselines:

- Full SFT: Train on all examples with direct supervision
- Random subset: Train on randomly selected subset matching our data budget
- CoT distillation only: Apply chain-of-thought distillation to all examples

Metrics: Accuracy on MMLU-Pro test set.

Main Results

Table 7.5 presents the main fine-tuning results.

Key findings:

1. Our complexity-aware approach achieves the highest accuracy while using only 19% of the training data.

Table 7.5: Accuracy comparison of fine-tuning approaches on MMLU-Pro test set.

Method	Data Used	Qwen-2.5-1.5B	Qwen-2.5-3B
Base model	–	0.28	0.35
Full SFT	100%	0.39	0.51
Random subset	19%	0.34	0.44
CoT distill (all)	100%	0.45	0.57
Complexity-aware (ours)	19%	**0.52**	**0.64**

2. Compared to full SFT, we improve accuracy from 0.39 to 0.52 (+33%) for Qwen-2.5-1.5B and from 0.51 to 0.64 (+25%) for Qwen-2.5-3B.

3. Even compared to full CoT distillation on all data, our targeted approach achieves higher accuracy with 81% less data.

Ablation Study

We ablate the key components of our approach:

Table 7.6: Ablation study on Qwen-2.5-1.5B.

Configuration	Accuracy
Full method	**0.52**
Without stratification (CoT all)	0.45
Without CoT (SFT all)	0.39
Easy examples only (SFT)	0.36
Hard examples only (CoT)	0.48

The ablation confirms that both stratification and the combination of SFT (for easy examples) and CoT distillation (for hard examples) are necessary for optimal performance.

Complexity Threshold Analysis

We analyze the effect of the complexity threshold τ on performance:

Table 7.7: Effect of complexity threshold (percentile of training data classified as "hard").

Hard Fraction	Easy (SFT)	Hard (CoT)	Accuracy
10%	90%	10%	0.47
20%	80%	20%	**0.52**
30%	70%	30%	0.51
50%	50%	50%	0.49

Optimal performance is achieved when approximately 20% of examples are classified as hard and receive CoT distillation.

7.4 Summary of Results

Across all three studies, we demonstrate that signals from the model's forward pass provide valuable information for addressing key challenges in LLM deployment:

1. **Uncertainty estimation**: Token-wise entropy patterns achieve ROC-AUC up to 0.83 for predicting answer correctness.
2. **Language control**: PCA-identified language directions enable 47–55% reduction in code-switching.
3. **Training efficiency**: Complexity-aware fine-tuning improves accuracy while using 81% less training data.

These results validate our central hypothesis that accessible signals from forward passes encode sufficient information to improve reliability, control, and efficiency without requiring architectural modifications.

Chapter 8
Discussion and conclusion

8.1 Summary of Contributions

This thesis investigated how signals from a single forward pass through large language models—both output distributions and internal representations—can be leveraged to address fundamental challenges in LLM deployment. We presented three interconnected studies that demonstrate the practical utility of this approach.

Study 1: Uncertainty Estimation via Token-Wise Entropy. We demonstrated that analyzing token-level entropy patterns during generation provides effective signals for predicting answer correctness. The max-to-min entropy ratio achieved ROC-AUC scores up to 0.83 on the MMLU-Pro benchmark, outperforming baseline confidence measures including first-token probability, mean sequence probability, and verbalized confidence. This work shows that the dynamics of uncertainty throughout generation—not just aggregate measures—contain valuable information about model reliability.

Study 2: Language Steering in Latent Space. We discovered that language-specific directions can be reliably identified in LLM latent space through simple PCA analysis, achieving 95–99% classification accuracy. By steering model activations along these directions during inference, we reduced unwanted code-switching by up to 55% as measured by KL divergence, without requiring additional training or significant degradation in generation quality. This demonstrates that behavioral control can be achieved through geometric manipulation of internal representations.

Study 3: Complexity-Aware Fine-Tuning. We developed a training pipeline that uses entropy to stratify training data by complexity and applies targeted interventions: direct supervised fine-tuning for easy examples and chain-of-thought distillation for hard examples. This approach achieved accuracy improvements from 0.39 to 0.52 (Qwen-2.5-1.5B) and 0.51 to 0.64 (Qwen-2.5-3B) while using 81% less training data than baseline methods. This shows that understanding input complexity through output distributions enables more efficient use of training resources.

8.2 Unifying Insights

The three studies share a common finding: the signals produced during a single forward pass through an LLM encode rich information that goes beyond the immediate task of next-token prediction. Specifically:

1. **Output distributions reflect model confidence**: High entropy indicates uncertainty, and the pattern of entropy changes throughout generation correlates with answer correctness.

2. **Internal representations encode interpretable structure**: Language identity is represented as linear directions in activation space, enabling geometric interventions for behavioral control.

3. **Entropy measures input complexity**: The model's uncertainty about a training example reflects its difficulty, enabling intelligent data stratification.

These insights suggest that LLMs develop internal representations of their own capabilities and limitations that can be accessed and exploited without explicit training.

8.3 Position in the Research Landscape

Our work contributes to several active research areas:

Uncertainty quantification: We extend prior work on semantic entropy [19] by demonstrating that token-wise entropy dynamics provide complementary signals. Our approach requires only a single forward pass, making it practical for real-time applications.

Interpretability and steering: Building on discoveries about linear structure in LLM representations [29, 36], we provide practical methods for language control that require no training and minimal computational overhead.

Efficient training: Our complexity-aware approach relates to curriculum learning [3] and data selection [34], but uniquely combines automatic complexity estimation with targeted training interventions.

8.4 Limitations

Several limitations should be acknowledged:

1. **Model dependence**: Our methods have been evaluated primarily on the Llama and Qwen model families. Generalization to other architectures (e.g., Mixture of Experts models) requires further validation.

2. **Language coverage**: The language steering experiments focused on English-Russian pairs. Extension to more diverse language combinations and low-resource languages remains future work.

3. **Task specificity**: Experiments were conducted primarily on multiple-choice question answering (MMLU-Pro). Performance on other task types (open-ended generation, reasoning, coding) needs investigation.

4. **Scaling behavior**: While we observed that larger models show better entropy-correctness calibration, the scaling behavior of all proposed methods requires systematic study.

8.5 Future Directions

This research opens several promising directions:

1. **Combining signals**: The three types of signals (entropy patterns, internal representations, complexity scores) could be combined into unified frameworks for model monitoring and control.

2. **Online adaptation**: The methods could be extended to enable online adaptation during deployment, automatically adjusting model behavior based on observed uncertainty or language patterns.

3. **Multi-dimensional steering**: Beyond language, the geometric structure of latent space could be exploited for steering other attributes such as formality, safety, or domain expertise.

4. **Theoretical foundations**: Developing theoretical understanding of why these signals are predictive would strengthen the foundations for future applications.
5. **Integration with alignment**: The uncertainty and control mechanisms developed here could be integrated with RLHF and other alignment techniques to improve model safety.

8.6 Conclusion

This thesis demonstrated that accessible signals from a single forward pass through large language models provide valuable information for addressing fundamental deployment challenges. Token-wise entropy patterns predict answer correctness, latent space geometry enables language control, and complexity-aware data stratification improves training efficiency. Together, these contributions establish that exploiting forward-pass signals offers a practical path toward more reliable, controllable, and efficient LLMs—without requiring architectural modifications or expensive retraining.

Acknowledgements

I would like to express my gratitude to my supervisor, Alexey Zaytsev, for his guidance and support throughout this research. I also thank my co-authors—Mikhail Sychev, Andrey Grishin, Danil Smorchkov, Ivan Molybog, and Egor Kondusov—for their valuable contributions to the published papers that form the basis of this thesis.

This research was conducted at Skolkovo Institute of Science and Technology (Skoltech). I acknowledge the computational resources provided by the Skoltech HPC cluster.

Innovations

Innovation component of your research project (if any), i.e.:

- Start-up potential
- Industrial application of research results

Bibliography

[1] Ahn, J., Oh, Y., Im, H., Yoon, S., and Cho, S. Do multilingual language models think better in english? *arXiv preprint arXiv:2308.01223* (2024).

[2] Belinkov, Y., Durrani, N., Dalvi, F., Sajjad, H., and Glass, J. What do neural machine translation models learn about morphology? *arXiv preprint arXiv:1704.03471* (2017).

[3] Bengio, Y., Louradour, J., Collobert, R., and Weston, J. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning* (2009), pp. 41–48.

[4] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in neural information processing systems 33* (2020), 1877–1901.

[5] Conneau, A., Kruszewski, G., Lample, G., Barrault, L., and Baroni, M. What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. *arXiv preprint arXiv:1805.01070* (2018).

[6] Elhage, N., Hume, T., Olsson, C., Schiefer, N., Henighan, T., Kravec, S., Hatfield-Dodds, Z., Lasenby, R., Drain, D., Chen, C., et al. Toy models of superposition. *arXiv preprint arXiv:2209.10652* (2022).

[7] Furlanello, T., Lipton, Z., Tschannen, M., Itti, L., and Anandkumar, A. Born again neural networks. *International Conference on Machine Learning* (2018), 1607–1616.

[8] Gal, Y., and Ghahramani, Z. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *International conference on machine learning* (2016), PMLR, pp. 1050–1059.

[9] Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. On calibration of modern neural networks. In *International conference on machine learning* (2017), PMLR, pp. 1321–1330.

[10] Ho, N., Schmid, L., and Yun, S.-Y. Large language models are reasoning teachers. *arXiv preprint arXiv:2212.10071* (2023).

[11] Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., Attariyan, M., and Gelly, S. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning* (2019), PMLR, pp. 2790–2799.

[12] Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685* (2022).

[13] Jawahar, G., Sagot, B., and Seddah, D. What does bert learn about the structure of language? *ACL 2019* (2019).

[14] Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y. J., Madotto, A., and Fung, P. Survey of hallucination in natural language generation. *ACM Computing Surveys 55*, 12 (2023), 1–38.

[15] Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., Casas, D. d. l., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., et al. Mistral 7b. *arXiv preprint arXiv:2310.06825* (2023).

[16] Jiang, Z., Araki, J., Ding, H., and Neubig, G. How can we know when language models know? on the calibration of language models for question answering. *Transactions of the Association for Computational Linguistics 9* (2021), 962–977.

[17] Kadavath, S., Conerly, T., Askell, A., Henighan, T., Drain, D., Perez, E., Schiefer, N., Hatfield-Dodds, Z., DasSarma, N., Tran-Johnson, E., et al. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221* (2022).

[18] Koh, P. W., and Liang, P. Understanding black-box predictions via influence functions. In *International conference on machine learning* (2017), PMLR, pp. 1885–1894.

[19] Kuhn, L., Gal, Y., and Farquhar, S. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. *arXiv preprint arXiv:2302.09664* (2023).

[20] Lakshminarayanan, B., Pritzel, A., and Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems 30* (2017).

[21] Li, K., Patel, O., Viégas, F., Pfister, H., and Wattenberg, M. Inference-time intervention: Eliciting truthful answers from a language model. *Advances in Neural Information Processing Systems 36* (2024).

[22] Libovický, J., Rosa, R., and Fraser, A. Language-neutral bert and what it understands. *arXiv preprint arXiv:2004.02070* (2020).

[23] Lin, S., Hilton, J., and Evans, O. Teaching models to express their uncertainty in words. *arXiv preprint arXiv:2205.14334* (2022).

[24] Magister, L. C., Mallinson, J., Adamek, J., Malmi, E., and Severyn, A. Teaching small language models to reason. *arXiv preprint arXiv:2212.08410* (2023).

[25] Malinin, A., and Gales, M. Uncertainty estimation in autoregressive structured prediction. *arXiv preprint arXiv:2002.07650* (2021).

[26] Mikolov, T., Yih, W.-t., and Zweig, G. Linguistic regularities in continuous space word representations. *Proceedings of NAACL-HLT* (2013), 746–751.

[27] Olah, C., Mordvintsev, A., and Schubert, L. Feature visualization. *Distill 2*, 11 (2017), e7.

[28] OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774* (2023).

[29] Park, K., Choe, Y. J., and Veitch, V. The linear representation hypothesis and the geometry of large language models. *arXiv preprint arXiv:2311.03658* (2023).

[30] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems 32* (2019).

[31] Pfeiffer, J., Rücklé, A., Poth, C., Kamath, A., Vulić, I., Ruder, S., Cho, K., and Gurevych, I. Adapterhub: A framework for adapting transformers. *arXiv preprint arXiv:2007.07779* (2020).

[32] Pires, T., Schlinger, E., and Garrette, D. How multilingual is multilingual bert? *arXiv preprint arXiv:1906.01502* (2019).

[33] Post, M. A call for clarity in reporting bleu scores. *arXiv preprint arXiv:1804.08771* (2018).

[34] Settles, B. Active learning literature survey. *University of Wisconsin-Madison Department of Computer Sciences* (2009).

[35] Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288* (2023).

[36] Turner, A. M., Thiergart, L., Udell, D., Leech, G., Mini, U., and MacDiarmid, M. Activation addition: Steering language models without optimization. *arXiv preprint arXiv:2308.10248* (2023).

[37] Wang, Y., Ma, X., Zhang, G., Ni, Y., Chandra, A., Guo, S., Ren, W., Arulraj, A., He, X., Jiang, Z., et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *arXiv preprint arXiv:2406.01574* (2024).

[38] Wendler, C., Veith, V., Haim, N., Kapur, R., Hassid, M., Pietrek, M., and Eichler, N. Do llamas work in english? on the latent language of multilingual transformers. *arXiv preprint arXiv:2402.10588* (2024).

[39] Winata, G. I., Madotto, A., Lin, Z., Liu, R., Yosinski, J., and Fung, P. Language models are few-shot multilingual learners. *arXiv preprint arXiv:2109.07684* (2021).

[40] Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., et al. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations* (2020), pp. 38–45.

[41] Wu, S., and Dredze, M. Beto, bentz, becas: The surprising cross-lingual effectiveness of bert. *arXiv preprint arXiv:1904.09077* (2019).

[42] Xie, Q., Luong, M.-T., Hovy, E., and Le, Q. V. Self-training with noisy student improves imagenet classification. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2020), 10687–10698.

[43] Yang, A., Yang, B., Hui, B., Zheng, B., Yu, B., Zhou, C., Li, C., Li, C., Liu, D., Huang, F., et al. Qwen2 technical report. *arXiv preprint arXiv:2407.10671* (2024).

Appendix