National Academy of Sciences of Ukraine
Institute of Mathematics

Qualifying scientific work
on the rights of the manuscript

SHAMRAI Maksym Borysovych

UDC 004.8:519.61:62-5

Low-rank approximations, perturbation bounds, and their role in model compression

113 Applied Mathematics
11 Mathematics and Statistics

Thesis
for the degree of Doctor of Philosophy

The thesis contains the results of own research. The use of ideas, results and texts of other authors have an appropriate citation.

_____ M.B. Shamrai

Supervisor
D.Sc., Academician of NAS of Ukraine
TYMOKHA Oleksandr Mykolayovych

Kyiv — 2026

Національна академія наук України
Інститут математики

Кваліфікаційна наукова
праця на правах рукопису

ШАМРАЙ Максим Борисович

УДК 004.8:519.61:62-5

Низькорангові наближення, оцінки збурень та їх роль у компресії моделей

113 Прикладна математика
11 Математика та статистика

Дисертація
на здобуття ступеня доктора філософії

Дисертація містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

_____ М.Б. Шамрай

Науковий керівник
доктор фіз.-мат. наук, академік НАН України
ТИМОХА Олександр Миколайович

Київ — 2026

Анотація

Шамрай М.Б. Низькорангові наближення, оцінки збурень та їх роль у компресії моделей. — Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття ступеня доктора філософії за спеціальністю 113 Прикладна математика. – Інститут математики НАН України, Київ, 2026.

Дисертацію присвячено одній із ключових задач сучасної прикладної математики та машинного навчання — розробленню методів стискання великих нейронних моделей із збереженням передбачуваної якості розв'язання прикладних задач. Із зростанням розмірності моделей їх практичне використання дедалі більше обмежується вимогами до пам'яті, затримки обчислень та енергоспоживання, тоді як більшість наявних схем компресії мають переважно евристичний характер. У дисертації запропоновано єдиний збурювальний підхід, у межах якого рішення щодо компресії спираються на явні кількісні гарантії, а не на ситуативні правила. Дослідження поєднує теорію матричних збурень, низькорангові наближення, емпіричний аналіз прунінгу великих мовних моделей, а також теоретичні гарантії робастності для задач керування та навчання з підкріпленням.

Метою дисертації є побудова математично обґрунтованих методів, які пов'язують параметричні збурення, спричинені компресією, з критеріями якості кінцевих прикладних задач. Об'єктом дослідження є низькорангові та розріджені представлення моделей машинного навчання. Предмет дослідження охоплює збурення сингулярних значень конкатенованих матриць, контрольоване за похибкою групування для спільної SVD-компресії, мовозалежні ефекти безнавчального прунінгу великих мовних моделей, побудову геометрії мов на основі сигналів важливості ваг, а також неасимптотичні оцінки деградації якості в задачах керування та навчання з підкріпленням за структурного розрідження параметрів.

У розділі 1 подано систематизований огляд наукової літератури — від класичних апроксимаційних підходів до сучасних глибоких нейронних моделей, великих мовних моделей і нейромережевих політик у задачах керування та навчання з підкріпленням. Проаналізовано основні парадигми компресії, зокрема низькорангові наближення та прунінг, а також їх теоретичні засади. Особливу увагу приділено відкритим математичним проблемам: одержанню інтерпретованих неасимптотичних оцінок збурень, формалізації умов доцільності спільної матричної компресії, а також установленню зв'язку між параметричними змінами після прунінгу та метриками якості й робастності. На цій основі сформульовано основний дослідницький розрив, який усувається в дисертації: відсутність цілісного конвеєра типу "теорія–практика", що одночасно забезпечував би строгі гарантії та придатні до застосування правила компресії.

У розділі 2 досліджено збурення сингулярних значень і низькорангові наближення конкатенованих матриць. Для блоків $A_i \in \mathbb{R}^{m\times n_i}$, $i = 1, \dots, k$, їх спільне подання має вигляд

$$M = [A_1\ A_2\ \cdots\ A_k].$$

Ключовим є питання, за яких умов матриці доцільно стискати окремо, а за яких спільно, використовуючи єдину низьковимірну базу. Для рангу r найкраще наближення задає явний критерій "хвостової енергії" для оцінювання похибки реконструкції. Для кількісного аналізу чутливості до блокових збурень $E = [E_1\ \dots\ E_k]$ використано оцінки сингулярних значень типу Вейля, а також побудовано структурно чутливі оцінки на основі матриць $M^\top M$ і $MM^\top$, що дає змогу встановити, у яких випадках кожне з цих подань є точнішим та більш інтерпретованим. Запропоновано правило групування з урахуванням збурень, згідно з яким кандидатне об'єднання приймається лише тоді, коли відносна похибка реконструкції менша за заданий поріг допустимої похибки. Таким чином, задачу групування переведено з евристичної площини у формалізовану задачу допустимості з явним сертифікатом об'єднання.

На основі критерію похибки апроксимації усіченим SVD та глобальних і інкрементних оцінок, виведених і доведених у цій дисертації, розроблено три стратегії кластеризації/стиснення: за максимум-нормою (швидка та консервативна), на основі залишків (найточніша та найнадійніша) та наближена інкрементна (масштабована для великих колекцій). Ці доведені глобальні та інкрементні оцінки забезпечують контроль похибки відновлення, що робить послідовні рішення щодо об'єднання теоретично обґрунтованими. Числові експерименти продемонстрували стійкий компроміс між швидкодією алгоритмів і "жорсткістю" гарантій. Для наборів даних рівень компресії зростає зі збільшенням допустимої похибки, тоді як залежність від цільового рангу виявляється немонотонною через узгодженість підпросторів між блоками. Додаткове порівняння з *random clustering*, *k-means* та *HDBSCAN* показало, що класичні геометричні цілі кластеризації недостатньо добре узгоджуються із задачею контрольованої конкатенованої SVD-компресії.

У розділі 3 прунінг розглянуто як структурне параметричне збурення. Для параметрів моделі Θ проріджені параметри задаються як $\widehat{\Theta} = \Theta + \delta\Theta$. У першому емпіричному дослідженні порівняно безнавчальні методи SparseGPT і Wanda для моделей LLaMA, LLaMA 2 та Mistral за неструктурованого та 2:4 напівструктурованого прунінгу на рівні 50%. Прунінг було відкалібровано на українському корпусі UberText 2.0 та зіставлено з калібруванням на англомовному корпусі c4 для аналізу впливу мовної невідповідності. Показано, що SparseGPT загалом є стабільнішим за Wanda щодо збереження якості, особливо в режимі 2:4, а невідповідність між мовою калібрування та мовою оцінювання призводить до погіршення результатів на українських текстах. Для 2:4 напівструктурованого прунінгу досягнуто зменшення обсягу пам'яті приблизно на 41% за збереження конкурентної перплексії.

Друге емпіричне дослідження присвячено побудові геометрії мов на основі сигналів важливості ваг, отриманих після прунінгу. Для кожної

мови L формується бінарний вектор важливості $z_L \in \{0,1\}^d$, а як метрика використовується відстань Геммінга. На основі багатомовних великих мовних моделей і великих текстових корпусів побудовано такі вектори для 106 мов. Отриманий метричний простір відтворює змістовну структуру мовних сімей і гілок, а також виявляє правдоподібні міжсімейні зв'язки, пов'язані з мовними контактами. Кластеризаційний аналіз демонструє краще узгодження з тоншими, гілковими мітками, ніж із макросімейною класифікацією. Водночас експерименти з *transfer learning* показують, що сама лише мовна відстань не гарантує стабільного поліпшення якості.

У теоретичній частині розділу 3 одержано явні гарантії робастності для OBD/OBS-подібного прунінгу багатошарових нейромережевих політик із 1-ліпшицевими функціями активації. Виведено замкнені пошарові оцінки збурення виходу та деградації якості керування, які обчислюються за характеристиками непрорідженої моделі та спектральними нормами. Для задач навчання з підкріпленням деградацію функціонала якості обмежено через повну варіацію та встановлено неасимптотичну сертифікацію, що забезпечує практичний механізм бюджетування прунінгу до його виконання та валідації після нього без необхідності обчислення глобального Гессіана.

Наукова новизна дисертації полягає в поєднанні в межах єдиного узгодженого підходу теорії збурень, компресійно-орієнтованого низькорангового групування, багатомовних експериментів із прунінгу, побудови геометрії мов на основі вагових сигналів, а також сертифікації якості для задач керування та навчання з підкріпленням. Практичне значення одержаних результатів визначається їх безпосередньою придатністю до трьох основних сценаріїв застосування: (i) низькорангової компресії колекцій матриць із контролем похибки; (ii) мовно-орієнтованого прунінгу великих мовних моделей; (iii) безпеко-орієнтованого прунінгу нейромережевих контролерів і політик навчання з підкріпленням. Запропоно-

вані методи забезпечують інтерпретовані та обчислювані критерії, які пов'язують кроки компресії з надійністю та якістю розв'язання прикладних задач.

Ключові слова: компресія моделей, низькорангове наближення, конкатеновані матриці, збурення сингулярних значень, усічений SVD, компресійно-орієнтована кластеризація, великі мовні моделі, прунінг, геометрія мов, робастне керування, навчання з підкріпленням, неасимптотичні оцінки, наближений розв'язок, межі, чисельний метод, алгоритм, складні динамічні системи, системи керування, нелінійні системи, замкнена система, негладка оптимізація, оптимальні функціонали, оптимальне керування, позитивно напіввизначені матриці, розріджені матриці, SVD, оцінка похибки.

Abstract

Shamrai M.B. Low-rank approximations, perturbation bounds, and their role in model compression. — Qualifying scientific work on the rights of the manuscript.

Thesis for the degree of Doctor of Philosophy, Speciality 113 Applied Mathematics. – Institute of Mathematics of NAS of Ukraine, Kyiv, 2026.

The dissertation addresses one of the key problems of modern applied mathematics and machine learning: how to compress large neural models while preserving predictable task-level quality. As model size grows, practical deployment is constrained by memory, latency, and energy budgets, and common compression pipelines remain heavily heuristic. This thesis develops a unified perturbation-based framework in which compression decisions are guided by explicit quantitative guarantees rather than ad hoc rules. The work combines matrix perturbation theory, low-rank approximation, empirical analysis of large language model pruning, and theoretical robustness guarantees for control and reinforcement learning.

The objective of the dissertation is to construct mathematically grounded methods that connect parameter perturbations caused by compression to downstream quality criteria. The research object is low-rank and sparse representations of machine-learning models. The research subject includes: singular-value perturbations of concatenated matrices, error-controlled grouping for joint SVD compression, language-dependent effects of training-free LLM pruning, pruning-derived language geometry, and nonasymptotic control/RL degradation bounds under structured parameter sparsification.

In Chapter 1, a systematic literature review is presented, covering the evolution from classical approximation methods to modern deep neural models, large language models, and neural policies in control and reinforcement learning. The chapter analyzes low-rank approximation and pruning as com-

pression paradigms and compares their theoretical foundations. Special attention is given to unresolved mathematical issues: deriving interpretable nonasymptotic perturbation bounds, formalizing when joint matrix compression is beneficial, and connecting pruning-induced parameter changes to downstream quality and robustness metrics. Based on this analysis, the chapter formulates the key research gap addressed in the dissertation: the absence of a consistent theory-to-practice pipeline that simultaneously provides rigorous guarantees and practically usable compression rules. This gap definition serves as the conceptual bridge from the review chapter to the original results of Chapters 2 and 3.

In Chapter 2, we study singular-value perturbations and low-rank approximation of concatenated matrices. Given blocks $A_i \in \mathbb{R}^{m\times n_i}$, $i = 1, \ldots, k$, their joint representation is

$$M = [A_1\ A_2\ \cdots\ A_k].$$

The central question is whether matrices should be compressed separately or jointly via a shared low-dimensional basis. For rank-r compression, the best approximation gives an explicit tail-energy criterion for reconstruction quality. To quantify sensitivity to blockwise perturbations $E = [E_1\ \ldots\ E_k]$, the analysis uses singular-value bounds in the Weyl form with additional structure-aware estimates derived from $M^\top M$ and $MM^\top$ to clarify when each viewpoint is tighter and more interpretable. A perturbation-aware grouping rule is then introduced: a candidate merge is accepted only if the relative reconstruction error is less than a user-specified tolerance. This transforms grouping from a heuristic procedure into a constrained feasibility problem with an explicit merge certificate.

From the truncated-SVD approximation-error criterion and the global/incremental bounds derived and proved in this dissertation, three clustering/compression strategies are developed: max-norm (fast and conservative), residual-based (most accurate and reliable), and approximate incremental (scalable in large collections). These proved global and incremental

error bounds certify reconstruction-error control, making sequential merge decisions theoretically justified. Numerical experiments show a consistent trade-off between runtime and guarantee tightness. Across datasets, compression increases as the admissible reconstruction tolerance is relaxed, while dependence on target rank is non-monotonic due to inter-block subspace alignment. Additional comparison with random clustering, k-means, and HDBSCAN confirms that classical geometric clustering objectives are poorly aligned with controlled concatenated-SVD compression.

In Chapter 3, pruning is analyzed as structured parameter perturbation. For model parameters Θ pruned parameters are $\widehat{\Theta} = \Theta + \delta\Theta$. In the first empirical study, training-free methods SparseGPT and Wanda are compared on LLaMA, LLaMA 2, and Mistral under unstructured and 2:4 semi-structured 50% sparsity. Pruning is calibrated on Ukrainian UberText 2.0 and contrasted with English c4 calibration to test language mismatch. The results show that SparseGPT is generally more stable than Wanda in quality retention, especially in 2:4 mode, and that calibration-language mismatch degrades Ukrainian evaluation quality. For 2:4 semi-structured pruning, memory footprint is reduced by approximately 41% while preserving competitive perplexity.

The second empirical study develops a new language-geometry methodology from pruning-derived weight saliency. For each language L, we construct binary weight-importance vectors $z_L \in \{0,1\}^d$, and measure distances using the Hamming distance. Using multilingual LLMs and large corpora, vectors are computed for 106 languages. The resulting geometry recovers meaningful family- and branch-level structure and also identifies plausible cross-family links related to contact effects. Clustering analysis demonstrates stronger alignment with finer-grained branch labels than with broad macro-family labels. At the same time, transfer-learning experiments indicate that language distance alone is not sufficient to guarantee consistent downstream gains.

The theoretical part of Chapter 3 derives explicit robustness guarantees for OBD/OBS-style pruning of multilayer neural policies with 1-Lipschitz activations. Closed-form layer-local bounds are obtained for output perturbation and control degradation, computable from unpruned model quantities and spectral norms. For reinforcement learning, return degradation is bounded via total variation, and nonasymptotic certification is established through a chain of bounds, providing a practical mechanism for pre-pruning budgeting and post-pruning validation without global Hessian computation.

The scientific novelty of the dissertation lies in combining perturbation analysis, compression-aware low-rank grouping, multilingual pruning experiments, language-geometry construction from weight-level signals, and control/RL certification in a single consistent framework. The practical significance is determined by direct applicability to three deployment scenarios: (i) error-controlled low-rank compression of matrix collections, (ii) language-aware pruning of large language models, and (iii) safety-oriented pruning of neural controllers and reinforcement-learning policies. The developed methods provide interpretable and computable criteria that connect compression actions to downstream reliability.

Key words: model compression, low-rank approximation, concatenated matrices, singular-value perturbation, truncated SVD, compression-aware clustering, large language models, pruning, language geometry, robust control, reinforcement learning, nonasymptotic bounds, approximate solution, bounds, numerical method, algorithm, complex dynamic systems, control systems, nonlinear systems, closed-loop system, non-smooth optimization, optimal functionals, optimal control, positive semi-definite matrices, sparse matrices, SVD, error estimation.

List of publications of PhD candidate

1. Shamrai M., Analysis of Perturbations of Singular Values in Concatenated Matrices, Ukrainian Mathematical Journal, 77, pp. 1136–1149, 2025,
https://doi.org/10.1007/s11253-025-02512-1. (Scopus – Q3, WoS – Q3, SJR – Q3).

2. Shamrai M., Closed-Form Robustness Bounds for Second-Order Pruning of Neural Controller Policies, Proceedings of the Institute of Applied Mathematics and Mechanics NAS of Ukraine, 39, pp. 81-89, 2025,
https://doi.org/10.37069/1683-4720-2025-39-7. (Category B journal).

3. Shamrai M., Nonasymptotic Bounds on Return Degradation for OBD-Pruned Neural Controllers. Bulletin of the Taras Shevchenko National University of Kyiv, Physics and Mathematics, 81(2), pp. 155-158, 2025, https://doi.org/10.17721/1812-5409.2025/2.24 (Scopus – Q4, SJR – Q4).

4. Shamrai M., Concatenated Matrix SVD: Compression Bounds, Incremental Approximation, and Error-Constrained Clustering, 2026,
2601.11626

5. Shamrai M., Hamolia V., Deep Language Geometry: Constructing a Metric Space from LLM Weights, In Proceedings of the 15th International Conference on Recent Advances in Natural Language Processing, pp. 1127–1136, Varna, Bulgaria, 2025,
https://aclanthology.org/2025.ranlp-1.130/ (Scopus – Q2)

6. Shamrai M., Language-Specific Pruning for Efficient Reduction of Large Language Models, In Proceedings of the Third Ukrainian Natural Language Processing Workshop (UNLP) @ LREC-COLING 2024, pp. 135–140, Torino, Italia. ELRA and ICCL, 2024, https://aclanthology.org/2024.unlp-1.16.pdf.

7. Shamrai M., Perturbation Analysis of Singular Values in Concatenated Matrices, Abstracts of the International Conference of Young Mathematicians, Kyiv, The Institute of Mathematics of the National Academy of Sciences of Ukraine, 2025, https://www.imath.kiev.ua/~young/youngconf2025/abstracts/Shamrai.pdf.

8. Shamrai M., Control Error Bound for Pruned Neural Controllers, VIII International Scientific Conference "Modern Problems of Mechanics", Kyiv, Taras Shevchenko National University of Kyiv, 2025, https://drive.google.com/file/d/1OTC7p6qjED_sRUVvGoo8UFQPVUGihQEe.

Acknowledgements

I express my sincere gratitude to my supervisor, Dr. Oleksandr M. Tymokha, for his guidance, insightful advice, and continuous support throughout my postgraduate studies. His thoughtful comments, patience, and encouragement have been invaluable in shaping this research. I am also grateful to Dr. Vyacheslav M. Boyko for proofreading the manuscript and for his insightful feedback during the department's seminars.

I would also like to express my deepest appreciation to the defenders of Ukraine for their courage and steadfastness in confronting the aggressor. Owing to their sacrifice and dedication, we have the opportunity to live, study, and work for the benefit of Ukraine.

This research was supported by grants from the Simons Foundation (SFI-PD-Ukraine-00014586, M.S.) and by project 0125U000299 of the National Academy of Sciences of Ukraine.

Contents

Notations

Unless stated otherwise, uppercase Latin letters denote matrices, while lowercase letters denote scalars or vectors according to context. Matrix and vector dimensions are inferred from context.

General linear-algebra notation

- $\mathbb{R}, \mathbb{N}$ — real numbers and positive integers.
- $\|\cdot\|_2, \|\cdot\|_F, \|\cdot\|_1$ — spectral (operator), Frobenius, and ℓ_1 norms.
- $\operatorname{rank}(A)$ — rank of matrix A.
- $\lambda_i(A)$ — ith eigenvalue (ordered nonincreasingly when relevant).
- $\sigma_i(A)$ — ith singular value (ordered nonincreasingly).
- $A^\top$ — matrix transpose.

Concatenated matrices and perturbations (Chapter 2)

- $A_i \in \mathbb{R}^{m \times n_i}$ — ith matrix block in a collection.
- $M = [A_1, \ldots, A_k]$ — horizontal concatenation of k blocks.
- $\widetilde{A}_i = A_i + E_i$ — perturbed block and its perturbation matrix E_i.
- $\widetilde{M} = M + E$ — perturbed concatenation, with $E = [E_1, \ldots, E_k]$.
- $M^\top M,\ MM^\top$ — Gram matrices used in spectral-perturbation analysis.
- A_r — best rank-r truncated SVD approximation of A.
- $\mathcal{E}_r(M)$ — optimal rank-r reconstruction error in Frobenius norm.
- $\widetilde{\sigma}_j(M)$ — incremental/truncated estimate of singular value $\sigma_j(M)$.

- Q_t, S_t — incremental orthonormal basis and reduced Gram matrix at step t.
- ε — relative reconstruction tolerance in compression-aware clustering.
- τ — absolute spectral budget for singular-value perturbation bounds.
- C (or $\mathcal{C}$), M_C — cluster index set and its concatenated matrix.

Pruning and language-geometry notation (Chapter 3)

- $W, \widehat{W}$ — dense and pruned weight matrices of a layer.
- X — layer input-activation matrix used for calibration.
- H — (local) Hessian or Hessian surrogate in second-order pruning.
- S_{ij} — weight-importance score used by SparseGPT/Wanda-type pruning rules.
- $\Theta = \{W_\ell, b_\ell\}_{\ell=1}^{L}$ — neural-policy parameter set across L affine layers.
- $\widehat{\Theta} = \Theta + \delta\Theta$ — pruned parameter set and induced parameter perturbation.
- δW_k — pruning perturbation of the kth weight matrix.
- $\pi(\cdot; \Theta)$ — neural policy (controller) parameterized by Θ.
- $B_\pi(\delta\Theta)$ — computable policy-output perturbation certificate.

Control and reinforcement-learning notation (Chapter 3)

- X, U — state and action spaces in deterministic nonlinear control setup.
- $\mathcal{S}, \mathcal{A}$ — finite state and action sets in MDP formulation.
- $P(s' \mid s, a)$ — transition kernel of the Markov decision process.
- $r(s, a), R_{\max}$ — reward function and uniform reward bound.

- $\gamma \in (0, 1)$ — discount factor.
- $J(\Theta), J(\pi)$ — expected discounted return (parameterized or policy form).
- d^{π} — discounted state-visitation distribution under policy π.
- $D_{\mathrm{TV}}(p, q), D_{\mathrm{KL}}(p\|q)$ — total-variation and Kullback–Leibler divergences.
- $m \in \{0, 1\}^d$ — binary pruning mask for a d-dimensional parameter vector.

Disambiguation conventions

- ϵ vs. ε — ϵ is used for small numerical thresholds (e.g., eigencutoffs), while ε denotes user-level reconstruction/error tolerances.

Introduction

Relevance of research topic. Model compression has become one of the central problems of modern applied mathematics and machine learning. The size of contemporary neural models makes their deployment difficult in real-world settings with limited memory, limited energy budget and strict latency constraints. Among the most effective compression directions are low-rank approximations and pruning. At the same time, practical pipelines in these directions are still largely heuristic, while their theoretical guarantees are often incomplete or too conservative for deployment-oriented decisions.

In low-rank compression, a key unresolved question is how to group multiple matrices for joint SVD compression with controlled reconstruction error. In pruning, a key unresolved question is how parameter perturbations induced by sparsification propagate to task-level quality, including language modeling quality, geometric structure extraction and control or reinforcement-learning objectives. Therefore, there is a strong need for a unified perturbation-based framework that combines rigorous bounds with practically applicable algorithms.

This thesis addresses exactly this gap. It develops spectral perturbation bounds for concatenated matrices, builds compression-aware clustering methods with explicit SVD error control, and studies pruning-induced perturbations both empirically (for multilingual large language models) and theoretically (for deterministic control and reinforcement learning). The topic is relevant both from the fundamental viewpoint of matrix perturbation theory and from the practical viewpoint of efficient and reliable model deployment.

Relation with academic programs, plans, themes, grants. This thesis was conducted at the Department of Mathematical Problems of Mechanics and Control Theory of the Institute of Mathematics of the Na-

tional Academy of Sciences of Ukraine as part of the research projects "Development and investigation of mathematical models of complex objects of mechanics and control systems" (2021–2025, state registration number 0121U100317), "Complex dynamical system in sciences: theory, mathematical modelling, numerical methods and implementation to advanced technology" (National Research Foundation of Ukraine, 2020-2024, state registration number 0120U104004), "Mathematical modeling of complex dynamical systems and processes actual to the state security" (2024-2025, state registration number 0123U100853). The research aligns with the priority areas of fundamental research established by the National Academy of Sciences of Ukraine in applied mathematics.

Purpose and objectives of research. *The purpose of the thesis* is to develop a mathematically grounded framework for analyzing perturbations in model compression, with focus on low-rank approximations of concatenated matrices and pruning of neural models, and to derive practical algorithms and guarantees for quality-preserving compression.

The research object is low-rank and sparse representations of machine-learning models, in particular concatenated matrix collections and pruned neural-network weights.

The research subject is given by: spectral perturbations of singular values for concatenated matrices, error-controlled compression-aware clustering, language-specific effects of training-free LLM pruning, construction of language geometry from pruning saliency, and nonasymptotic robustness and return-degradation bounds for OBD-pruned neural controllers.

Research methods. The dissertation uses methods of matrix analysis and numerical linear algebra, including singular value decomposition, Weyl-type perturbation inequalities, and the Eckart–Young–Mirsky theorem; methods of optimization and incremental low-rank updates; methods of statistical and computational experiments for large language models; and methods of control theory and reinforcement learning, including Lipschitz perturbation analysis and total-variation-based performance bounds.

Scientific novelty of the obtained results. The main results that determine the scientific novelty of the thesis and are submitted for its defense are the following:

1. Upper bounds were derived for singular-value perturbations of concatenated matrices under blockwise perturbations, with explicit comparison of Gram-matrix formulations based on $M^\top M$ and $MM^\top$, which clarifies when each viewpoint provides a tighter and more interpretable estimate.

2. A compression-aware formulation of matrix grouping was developed, where candidate groups are accepted only if predicted truncated-SVD reconstruction error satisfies a user-specified feasibility constraint.

3. Global and incremental bounds were obtained that connect singular-value growth with reconstruction-error control, enabling theoretically justified merge decisions in sequential matrix grouping.

4. Three clustering strategies for concatenated SVD compression were constructed and analyzed: max-norm (fast conservative), residual-based (provably accurate), and approximate incremental (scalable high-compression).

5. For LLM pruning, language dependence of calibration data was systematically quantified: at fixed sparsity budgets, SparseGPT demonstrates more stable quality retention than Wanda, especially in 2:4 semi-structured pruning and in Ukrainian-language evaluation.

6. A new approach to language geometry was proposed, where binary weight-importance vectors induced by pruning saliency define a metric space of languages. The approach was applied to 106 languages and validated by meaningful family-level clustering patterns.

7. For deterministic nonlinear control, closed-form layer-local robustness bounds were derived for OBD/OBS-style pruning, including additive multi-layer extensions computable from unpruned network quantities.

8. For reinforcement learning, nonasymptotic return-degradation bounds were derived for OBD-pruned policies, linking parameter perturbations to policy divergence in total variation and then to guaranteed return loss. This yields a practical pre-pruning budgeting and post-pruning validation pipeline.

Practical significance of the obtained results. The obtained results provide practical tools for quality-controlled compression of modern models. The developed bounds and algorithms can be used in: (i) grouping and compressing large matrix collections under explicit SVD error budgets, (ii) language-aware pruning of large language models, and (iii) safety-oriented pruning of neural controllers and reinforcement-learning policies. The proposed methods are applicable in scientific computing and engineering systems where memory efficiency must be combined with reliability guarantees.

Personal contribution of the PhD candidate. All core results presented in the dissertation were obtained by the PhD candidate. The candidate performed theoretical derivations, algorithm design, implementation and numerical experiments. In the co-authored publication with V. Hamolia, the main methodology and experimental analysis relevant to this dissertation were carried out by the PhD candidate.

Approbation of the thesis results. The main results of the thesis were reported and discussed at:

- Third Ukrainian Natural Language Processing Workshop (UNLP) at LREC-COLING 2024 (Torino, Italy, 2024);
- International Conference of Young Mathematicians (Kyiv, Institute of Mathematics of NAS of Ukraine, 2025);
- VIII International Scientific Conference “Modern Problems of Mechanics” (Kyiv, Taras Shevchenko National University of Kyiv, 2025);
- 15th International Conference on Recent Advances in Natural Language Processing (RANLP 2025, Varna, Bulgaria, 2025);

- Seminar of Young Scientists (Kyiv, Institute of Mathematics of NAS of Ukraine, 2026);

Publications. The results of the thesis were published in eight scientific publications. They include three journal articles (two indexed in Scopus, Q3 and Q4, and one published in a Category B journal), two full papers in international conference/workshop proceedings (one indexed in Scopus, Q2), two conference abstracts, and one preprint presenting extended results on concatenated-matrix SVD compression.

Structure and volume of thesis. The thesis contains annotations in Ukrainian and English, a list of the author's publications, acknowledgments, notations, an introduction, a literature review, three chapters, a conclusion, a list of references and three appendices. The total volume of the thesis is 156 pages. The list of references contains 120 items.

Chapter 1

Literature review

This chapter provides an introductory literature review for the dissertation. The chapter explains why neural-network compression is both practically necessary and mathematically rich, and it connects four core themes of the thesis: neural networks, large language models, neural policies, and compression methods.

Modern machine learning is shaped by a central tension. On one hand, very large models provide state-of-the-art performance in language, vision, and control [1–3]. On the other hand, these same models are difficult to deploy due to memory limits, latency constraints, and energy cost [4, 5]. As a consequence, compression is no longer a secondary optimization trick; it is a primary scientific and engineering problem [4, 6]. At the same time, many practical compression pipelines are still heuristic, while guarantees that are needed by scientific computing and control applications remain incomplete [7]. This motivates the perturbation-oriented viewpoint used throughout this dissertation.

1.1. Deep neural networks

Neural networks can be viewed as high-dimensional parametric approximators [8, 9]. Given an input vector $x \in \mathbb{R}^d$, a network with parameters Θ defines a map

$$f(x;\Theta)\colon \mathbb{R}^d \to \mathbb{R}^p.$$

In supervised learning, Θ is obtained by minimizing empirical risk [10]

$$\min_{\Theta} \frac{1}{N} \sum_{i=1}^{N} \ell\big(f(x_i;\Theta), y_i\big),$$

where ℓ is a task-specific loss. In modern practice, this optimization is high-dimensional, nonconvex, and data-intensive [3].

The mathematical perspective is useful here. Even when optimization is nonconvex, successful training often produces highly accurate models. This empirical success led to a major research direction: understanding the interplay between approximation power, optimization dynamics, generalization, and architecture design [11, 12]. The resulting ecosystem of methods now supports very large-scale representation learning, including language models and decision-making policies [1, 2].

A key practical observation is that contemporary models are often overparameterized [13]. Overparameterization supports optimization and expressivity, but it also introduces substantial redundancy. This redundancy is exactly what makes compression possible [4, 5]. Therefore, compression is not external to learning; it is deeply connected to model structure and training dynamics.

Most neural architectures are compositions of affine maps and pointwise nonlinearities. For a single layer,

$$h = \sigma(Wx + b),$$

where W is a weight matrix, b is a bias vector, and σ is an activation function. By composing such layers, one obtains deep mappings with rich nonlinear structure. Activation design (e.g., ReLU-like nonlinearities) is central for trainability and stable gradient propagation [14–18].

For this dissertation, three properties of neural networks are especially important:

1. **Matrix-structured parametrization.** Large parts of modern networks are linear operators represented by matrices. This enables low-rank and spectral analysis.

2. **Sensitivity to parameter perturbations.** Any compression method modifies parameters, and therefore induces output perturbations. Quantifying this effect is a core mathematical challenge.

3. **Layerwise heterogeneity.** Different layers respond differently to compression. This motivates local criteria (layerwise saliency, local Hessian surrogates, local spectral decay).

Classical feed-forward multilayer perceptrons were followed by specialized architectures: convolutional networks for spatial structure [19], recurrent networks for sequential data [20], and attention-based models for long-range dependencies [1]. The attention paradigm eventually became dominant in language modeling and many multimodal tasks [2, 21].

The rise of representation learning shifted the objective from narrow task-specific fitting to broad pretraining followed by adaptation [22, 23]. In this setting, models are trained on very large corpora and then reused across tasks. This paradigm improves transfer performance but further increases model size, making compression and efficient deployment indispensable [4, 5].

1.1.1. Neural policies in control and reinforcement learning. Neural policies provide a common mathematical language for nonlinear control and reinforcement learning (RL). They are central in robotics and autonomous systems, where one must optimize long-horizon behavior while preserving robust closed-loop performance. Recent progress spans deep RL benchmarks, large policy models, and vision-language-action systems [24, 25]. The goal of this section is twofold: to formalize the control–RL equivalence and to prepare a perturbation-based view of policy compression used later in the dissertation. Throughout this section, $t \in \mathbb{N}_0 = \{0, 1, 2, \dots\}$, all policies are assumed measurable, and discounted sums are assumed well-defined (for example, under bounded stage costs/rewards).

Nonlinear control formulation. We first formalize nonlinear stochastic control in discrete time, following standard treatments in dynamic programming and Markov decision processes [26, 27].

Definition 1.1 (Discrete-time nonlinear control system)**.** Let $\mathcal{X} \subseteq \mathbb{R}^{n_x}$ be the state space, $\mathcal{U} \subseteq \mathbb{R}^{n_u}$ be the action space, and Ξ be a disturbance space. A stochastic controlled system is

$$x_{t+1} = f(x_t, u_t, \xi_t), \qquad x_0 \sim \mu_0,$$

where $x_t \in \mathcal{X}$, $u_t \in \mathcal{U}$, $f \colon \mathcal{X} \times \mathcal{U} \times \Xi \to \mathcal{X}$, and $\xi_t \in \Xi$ is exogenous noise. Equivalently, the dynamics define a transition kernel

$$P(B \mid x, u) = \mathbb{P}\big(f(x, u, \xi) \in B\big), \qquad B \in \mathcal{B}(\mathcal{X}),$$

where $\mathcal{B}(\mathcal{X})$ is the Borel σ-algebra on $\mathcal{X}$ [27].

Definition 1.2 (Feedback controller and closed loop)**.** A feedback controller is a map $\kappa : \mathcal{X} \to \mathcal{U}$ (deterministic) or a conditional distribution $\kappa(\cdot \mid x)$ (stochastic). The closed-loop trajectory is generated by $u_t = \kappa(x_t)$ (or $u_t \sim \kappa(\cdot \mid x_t)$) and the dynamics above. For constrained control, one also requires $u_t \in \mathcal{U}(x_t)$ almost surely [28].

Definition 1.3 (Infinite-horizon discounted control objective)**.** Given stage cost $c : \mathcal{X} \times \mathcal{U} \to \mathbb{R}$ and discount $\gamma \in (0, 1)$, the value function of κ is

$$V^{\kappa}(x) = \mathbb{E}\Big[\sum_{t=0}^{\infty} \gamma^t c(x_t, u_t) \,\Big|\, x_0 = x\Big],$$

and aggregate performance is

$$J_{\mathrm{ctrl}}(\kappa) = \mathbb{E}_{x_0 \sim \mu_0}\big[V^{\kappa}(x_0)\big].$$

Here expectations are taken over trajectories induced by (μ_0, P, κ). The optimal control problem is

$$V^*(x) = \inf_{\kappa} V^{\kappa}(x), \qquad \kappa^* \in \arg\inf_{\kappa} J_{\mathrm{ctrl}}(\kappa).$$

Under standard regularity assumptions, V^* satisfies the Bellman optimality equation [26]

$$V^*(x) = \inf_{u \in \mathcal{U}(x)} \Big\{ c(x,u) + \gamma\, \mathbb{E}_\xi \big[V^*(f(x,u,\xi)) \big] \Big\},$$

where $\mathcal{U}(x)$ is the admissible-action set at state x.

Reinforcement learning as an MDP problem. The RL formulation uses the same dynamical skeleton, but maximizes rewards. To keep notation transparent, we use (x,u) in the control view and (s,a) in the MDP view; the identification between them is stated explicitly in the next subsection [27,29].

Definition 1.4 (Markov decision process)**.** An MDP is a tuple

$$\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, \gamma, \mu_0),$$

where $\mathcal{S}$ is the (measurable) state space, $\mathcal{A}$ is the (measurable) action space, $P(\cdot \mid s,a) \in \mathcal{P}(\mathcal{S})$ is the transition kernel, $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function, $\gamma \in (0,1)$ is the discount factor, and $\mu_0 \in \mathcal{P}(\mathcal{S})$ is the initial-state distribution [27].

Definition 1.5 (Policy, return, and value functions)**.** A policy π is a conditional distribution $\pi(a \mid s)$. For finite trajectory $\tau_{0:T} = (s_0, a_0, s_1, a_1, \ldots, s_T)$, the induced path density/mass is

$$p_\pi(\tau_{0:T}) = \mu_0(s_0) \prod_{t=0}^{T-1} \pi(a_t \mid s_t) P(s_{t+1} \mid s_t, a_t).$$

The RL objective is

$$J_{\mathrm{RL}}(\pi) = \mathbb{E}_{\pi,P}\Big[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \Big],$$

where $s_t \in \mathcal{S}$, $a_t \in \mathcal{A}$, and typically $|r(s,a)| \le R_{\max} < \infty$ [29]. Associated value functions are

$$V^\pi(s) = \mathbb{E}_{\pi,P}\Big[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \,\Big|\, s_0 = s \Big],$$

$$Q^\pi(s,a) = \mathbb{E}_{\pi,P}\Big[\sum_{t=0}^{\infty} \gamma^t r(s_t,a_t) \,\Big|\, s_0 = s, a_0 = a\Big],$$

with advantage $A^\pi(s,a) = Q^\pi(s,a) - V^\pi(s)$. The Bellman expectation equations are [26]

$$V^\pi(s) = \mathbb{E}_{a\sim\pi(\cdot|s),\, s'\sim P(\cdot|s,a)}\big[r(s,a) + \gamma V^\pi(s')\big],$$

$$Q^\pi(s,a) = \mathbb{E}_{s'\sim P(\cdot|s,a),\, a'\sim\pi(\cdot|s')}\big[r(s,a) + \gamma Q^\pi(s',a')\big],$$

and the optimal value satisfies

$$V^*(s) = \max_{a\in\mathcal{A}} \mathbb{E}_{s'\sim P(\cdot|s,a)}\big[r(s,a) + \gamma V^*(s')\big].$$

Neural controllers as the bridge between control and RL. The control and RL views are equivalent under the identifications $\mathcal{S} \equiv \mathcal{X}$, $\mathcal{A} \equiv \mathcal{U}$, $P(\cdot \mid s,a)$ induced by $f(s,a,\xi)$, and $r(s,a) = -c(s,a)$. Thus minimizing control cost is equivalent to maximizing RL return. This equivalence is central for compression analysis, because any compressed controller can be interpreted as a perturbed policy.

Definition 1.6 (Neural controller)**.** A neural controller is a parameterized policy. In deterministic form,

$$u_t = \mu_\Theta(x_t),$$

and in stochastic form,

$$a_t \sim \pi_\Theta(\cdot \mid s_t),$$

where $\mu_\Theta : \mathcal{X} \to \mathcal{U}$ and $\pi_\Theta : \mathcal{S} \to \mathcal{P}(\mathcal{A})$, with $\mathcal{P}(\mathcal{A})$ the set of probability measures on $\mathcal{A}$.

To connect directly with compression, let $\widehat{\Theta} = \Theta + \Delta\Theta$ denote compressed parameters (or more generally $\widehat{\Theta} = \mathcal{C}(\Theta)$ for a compression operator $\mathcal{C}$). Then compression quality can be expressed through induced performance gaps, for example

$$\Delta J_{\mathrm{ctrl}} = J_{\mathrm{ctrl}}(\kappa_{\widehat{\Theta}}) - J_{\mathrm{ctrl}}(\kappa_\Theta), \qquad \Delta J_{\mathrm{RL}} = J_{\mathrm{RL}}(\pi_\Theta) - J_{\mathrm{RL}}(\pi_{\widehat{\Theta}}).$$

The dissertation objective is to control these gaps using interpretable perturbation criteria.

Because perturbations propagate through feedback, small parameter changes may accumulate over long horizons. Therefore compression of neural policies is not only a size-reduction problem; it is a robust closed-loop performance problem, connecting modern compression methods with certifiable control-oriented guarantees [30–33].

1.1.2. Large language models. This subsection has three goals: to formalize the probabilistic training objective, to expose the matrix-centric Transformer structure relevant for compression, and to connect post-training alignment with policy optimization.

Autoregressive objective and perplexity. Large language models (LLMs) are neural sequence models trained to predict the next token from context [1,34]. For a tokenized corpus $\mathcal{D} = \{x^{(n)}_{1:T_n}\}_{n=1}^{N}$, the standard autoregressive objective is

$$\max_{\Theta} \sum_{n=1}^{N} \sum_{t=1}^{T_n} \log p_\Theta(x_t^{(n)} \mid x_{<t}^{(n)}).$$

where N is the number of training sequences, T_n is the length of sequence n, $x_t^{(n)}$ is token t in sequence n, and Θ denotes all trainable parameters. Equivalently, one minimizes token-level negative log-likelihood,

$$\mathcal{L}_{\mathrm{NLL}}(\Theta) = -\frac{1}{\sum_{n=1}^{N} T_n} \sum_{n=1}^{N} \sum_{t=1}^{T_n} \log p_\Theta(x_t^{(n)} \mid x_{<t}^{(n)}), \qquad \mathrm{PPL} = \exp\big(\mathcal{L}_{\mathrm{NLL}}\big),$$

where $\mathcal{L}_{\mathrm{NLL}}$ is average token-level cross-entropy and PPL is perplexity [11]. If $h_t^{(L)}$ is the final-layer hidden state at position t, then

$$p_\Theta(x_t \mid x_{<t}) = \mathrm{softmax}(W_{\mathrm{out}} h_t^{(L)} + b_{\mathrm{out}})_{x_t}.$$

where W_{out} and b_{out} are output projection parameters and the subscript x_t selects probability mass assigned to the true token.

Although the objective is simple, at scale it yields models that capture broad syntactic, semantic, and procedural regularities [2, 35]. Prominent modern examples include the LLaMA family, Mistral-family models, and recent open foundation models [36–40].

Transformer structure in matrix form. From a mathematical viewpoint, the Transformer architecture uses repeated linear projections, attention operations, and nonlinear feed-forward blocks [1]. For a layer input $H \in \mathbb{R}^{n\times d}$,

$$Q = HW_Q, \qquad K = HW_K, \qquad V = HW_V,$$

where n is sequence length, d is hidden width, and (W_Q, W_K, W_V) are learned projection matrices. Scaled dot-product attention is

$$\mathrm{Attn}(Q, K, V) = \mathrm{softmax}\left(\frac{QK^\top}{\sqrt{d_k}} + M\right) V,$$

where d_k is key/query head dimension and M is the causal mask. Multi-head attention is

$$\mathrm{MHA}(H) = \mathrm{Concat}(\mathrm{Attn}_1, \ldots, \mathrm{Attn}_m)W_O.$$

With residual connections and normalization, a standard block can be written as

$$\widetilde{H} = \mathrm{LN}(H + \mathrm{MHA}(H)), \qquad H^+ = \mathrm{LN}(\widetilde{H} + \mathrm{FFN}(\widetilde{H})),$$

where $\mathrm{LN}(\cdot)$ denotes layer normalization [41] and $\mathrm{FFN}(\cdot)$ is the position-wise feed-forward module.

Hence, despite algorithmic complexity, the architecture remains strongly matrix-centric. This viewpoint is central to modern compression methods, where low-rank structure and parameter sparsity are exploited in attention and feed-forward projections [6, 7].

Practical bottlenecks and motivation for compression. The practical influence of LLMs is now substantial. They are used for question answering, translation, code generation, scientific writing support, and multilingual information access [2, 21, 36]. However, this success reveals several bottlenecks:

- **Memory footprint:** multi-billion-parameter models are difficult to run on commodity hardware [4, 42].
- **Inference cost:** autoregressive decoding is latency-sensitive and expensive [35, 43].
- **Data and language imbalance:** low-resource languages often receive weaker representation [44].
- **Deployment gap:** server-grade models cannot always be used in on-device or privacy-sensitive contexts [45].

These bottlenecks directly justify the empirical part of this dissertation, where pruning of multilingual LLMs is studied under explicit perturbation and quality criteria. Prior work already established strong baselines for post-training compression, including quantization and training-free pruning [7, 42, 45–47]. Yet language-specific effects and mathematically interpretable robustness guarantees remain active research topics.

1.2. Low-rank approximation and matrix factorization

In this dissertation, low-rank approximation is treated first as a *compression mechanism* for large matrix parameters. If $A \in \mathbb{R}^{m\times n}$ is stored densely, it requires mn scalars, whereas a rank-r factorization $A \approx BC$ with $B \in \mathbb{R}^{m\times r}$ and $C \in \mathbb{R}^{r\times n}$ uses $r(m+n)$ scalars. Thus, when $r \ll \min\{m, n\}$, low-rank structure yields substantial storage reduction, and matrix-vector products can be executed as two thin multiplies instead of one dense multiply.

The central theoretical tool is singular value decomposition (SVD), whose optimality for rank-constrained compression is classically described by the Eckart–Young–Mirsky framework [48–50]. For a matrix A, the rank-r compressed surrogate is

$$A \approx A_r = U_r \Sigma_r V_r^\top,$$

and its approximation loss is governed by the singular-value tail. We repeatedly use the classical result that this truncated SVD is the best rank-r approximation in Frobenius norm.

Theorem 1.7 (Eckart–Young–Mirsky [49,50]). *Let $A \in \mathbb{R}^{m\times n}$ have singular values $\sigma_1(A) \geq \sigma_2(A) \geq \cdots \geq 0$, and let*

$$A = U\Sigma V^\top, \qquad \Sigma = \operatorname{diag}(\sigma_1, \ldots, \sigma_p), \quad p = \min\{m, n\}.$$

For $r \in \{0, \ldots, p\}$, define the rank-r truncated SVD

$$A_r := U \begin{bmatrix} \operatorname{diag}(\sigma_1, \ldots, \sigma_r) & 0 \\ 0 & 0 \end{bmatrix} V^\top.$$

Then A_r is the best rank-r approximation to A in the Frobenius norm:

$$\|A - A_r\|_F = \min_{\operatorname{rank}(X)\leq r} \|A - X\|_F = \left(\sum_{j>r} \sigma_j(A)^2 \right)^{1/2}.$$

This theorem underlies all of our compression error formulas.

For low-rank compression, it is often convenient to analyze the symmetric positive semidefinite Gram matrix instead of the original rectangular matrix. This converts singular-value questions into eigenvalue questions and enables clean perturbation arguments in later sections. The next classical proposition [51] gives this exact spectral correspondence.

Proposition 1.8 ([51]). *For any matrix $A \in \mathbb{R}^{n\times m}$ with rank r, the singular values of A are given by*

$$\sigma_i(A) = \sqrt{\lambda_i(A^\top A)}, \quad 1 \leq i \leq r.$$

In this dissertation, compression is modeled as a structured perturbation of matrices and operators, so perturbation theory is not auxiliary but central to the analysis. It provides the mechanism for converting parameter perturbations into quantitative statements about spectral drift, approximation error, and stability of compressed models. Accordingly, we repeatedly use classical results of matrix analysis as the technical backbone of the theory developed below.

The first such tool is Weyl's monotonicity theorem, which formalizes how the eigenvalues of a Hermitian matrix move under additive perturbations. In the special case of positive semidefinite updates, it yields one-sided spectral control: ordered eigenvalues cannot decrease. Combined with Proposition 1.8, this immediately gives monotonic behavior of ordered singular values under block concatenation.

Theorem 1.9 (Weyl Monotonicity; cf. Corollary 4.9 in [52]). *Let $A, E \in \mathbb{R}^{n\times n}$ be Hermitian and write their eigenvalues in nonincreasing order,*

$$\lambda_1(A) \geq \cdots \geq \lambda_n(A), \qquad \lambda_1(E) \geq \cdots \geq \lambda_n(E).$$

Let $\tilde{\lambda}_1 \geq \cdots \geq \tilde{\lambda}_n$ be the eigenvalues of $A + E$. Then, for all $i = 1, \ldots, n$,

$$\tilde{\lambda}_i \in \big[\lambda_i(A) + \lambda_n(E),\ \lambda_i(A) + \lambda_1(E)\big].$$

In particular, if $E \succeq 0$, then

$$\tilde{\lambda}_i \geq \lambda_i(A), \qquad i = 1, \ldots, n.$$

The following is the Weyl's inequality [53], which gives a uniform bound on the eigenvalues of a symmetric matrix under perturbations.

Theorem 1.10 (Weyl's inequality [53]). *Let $A, B \in \mathbb{R}^{n\times n}$ be symmetric matrices and $E = B - A$ is a perturbation. Then for every eigenvalue (ordered in any fixed manner),*

$$|\lambda_i(B) - \lambda_i(A)| \leq \|E\|_2.$$

Weyl's inequality ensures that if the perturbation E is small in spectral norm, then the eigenvalues of the perturbed matrix B remain close to those of the original matrix A. We will use this principle to study how the singular values of a concatenated matrix change under blockwise perturbations.

1.2.1. Concatenated matrices. Truncated singular value decomposition (SVD) is a fundamental tool for matrix compression, providing the optimal low-rank approximation of a matrix in the Frobenius norm [6, 49, 50, 54]. By representing a matrix through a small number of dominant singular vectors, truncated SVD enables compact storage, noise reduction, and efficient downstream computation. A common and long-standing extension in many domains is to horizontally concatenate matrices and apply a single truncated SVD to the resulting matrix. This idea appears, for example, in principal component analysis applied to stacked data matrices [54], as well as in the method of snapshots for model reduction [55]. This yields a shared low-rank factorization across all blocks, enabling direct reconstruction of the original matrices without higher-order tensor contractions or complex decoding procedures.

Concatenated SVD has been successfully applied across a broad range of domains. In large language models, joint SVD of concatenated weight matrices has been used to share low-rank projections across attention components, layers, or experts, enabling parameter reduction while preserving accuracy. Representative examples include unified QKV decompositions [56], intra-layer shared projections [57], cross-layer parameter sharing [58], and expert merging in mixture-of-experts architectures [59, 60]. In these settings, concatenation is typically guided by architectural structure (e.g., matrices belonging to the same layer or module) or semantic similarity (e.g., adjacent layers or related experts). Related ideas also appear in wireless signal processing, where concatenated SVD is used to design shared precoders across frequency bands [61], as well as in neuroscience and genomics, where large collections

of measurements are concatenated to obtain global low-dimensional representations across sessions, experimental conditions, or chromosomes [62, 63]. Across these application areas, concatenated SVD serves as a powerful tool for extracting shared structure from collections of matrices.

Despite its empirical success, existing uses of concatenated SVD rely on *predefined or heuristic grouping* of matrices. The decision of which matrices should share a low-rank basis is typically made manually based on domain knowledge, architectural constraints, or simple similarity measures.

However, such empirical grouping strategies do not provide guarantees on the resulting reconstruction error. In particular, concatenating matrices based on heuristic similarity or architectural proximity can lead to failure cases: when matrices are not well aligned, their joint representation can exhibit substantially higher effective rank, resulting in significantly larger approximation error than compressing them independently. This makes the choice of which matrices to concatenate a critical component of the compression pipeline, rather than a purely heuristic design decision. Consequently, one must determine *which subsets of matrices should be concatenated* so that joint compression improves parameter efficiency while still satisfying a prescribed reconstruction error or compression target. Crucially, existing approaches do not provide a principled mechanism for making this decision. As a result, compression quality depends heavily on ad hoc design choices, and there are no guarantees that merging additional matrices will not violate reconstruction constraints.

In contrast, the present work formulates matrix grouping as a *compression-driven clustering problem*. Rather than clustering matrices based on ambient-space distances or semantic heuristics, cluster formation is governed directly by predicted low-rank approximation error of the concatenated matrix. This enables principled selection of matrix groups under explicit reconstruction error budgets.

We also study the singular values of concatenated matrices and their perturbations under blockwise updates. This is important because practical compression pipelines often append blocks incrementally, and the quality of a shared low-rank basis is governed by how the leading singular values evolve during this process. In this dissertation, we extend classical perturbation theory from the single-matrix setting to concatenated matrices with blockwise perturbations. This extension makes it possible to track spectral variation under block concatenation and to derive guarantees tailored to incremental compression workflows. Proposition 1.8 allows us to study singular values through the eigenvalues of Gram matrices.

Let $M_t = [A_1, \ldots, A_t]$ and $M_{t+1} = [A_1, \ldots, A_t, A_{t+1}]$. Then

$$M_{t+1}M_{t+1}^\top = M_t M_t^\top + A_{t+1}A_{t+1}^\top,$$

and the update term $A_{t+1}A_{t+1}^\top$ is positive semidefinite. Therefore, by Weyl monotonicity for Hermitian matrices, appending a block cannot decrease the ordered eigenvalues of the Gram matrix, and hence cannot decrease the ordered singular values of the concatenated matrix.

1.2.2. Incremental representation of concatenated matrices. Incremental estimation of dominant singular values and singular subspaces has a long history in numerical linear algebra and machine learning. Early work on incremental principal component analysis and online SVD [64, 65] describes how to update covariance eigenbases as new samples arrive. Brand's incremental SVD algorithms [66, 67] extend these ideas to rank-one and block updates of the thin SVD, directly covering the case of appending new columns, which is equivalent to horizontal concatenation. Streaming PCA and subspace tracking methods [68, 69] maintain approximate dominant invariant subspaces under stochastic or adversarial updates. Randomized low-rank approximation techniques [6, 70] further improve scalability by maintaining approximate bases via random projections and periodic truncation.

The incremental truncated SVD estimator used in this work follows a classical design pattern and does not constitute a new SVD algorithm. Its update mechanism is equivalent to well-known incremental PCA and block-update SVD methods. The novelty of this work lies in how such estimators are used: we connect incremental singular value tracking to *explicit reconstruction-error control for concatenated matrices*, and embed it into clustering procedures whose merge decisions are driven directly by predicted low-rank approximation error. To our knowledge, existing incremental SVD and streaming PCA methods are not used to guide clustering or grouping under explicit SVD compression constraints.

Let $M_t = [A_1, \dots, A_t]$ denote the concatenation of t blocks. When a new block A_{t+1} is appended, the Gram matrix updates as

$$M_{t+1}M_{t+1}^\top = M_t M_t^\top + A_{t+1}A_{t+1}^\top,$$

which is a rank-$\leq n_{t+1}$ positive semidefinite perturbation. Instead of storing the full $m \times m$ matrix $M_t M_t^\top$, we maintain an orthonormal basis Q_t for the current column space of M_t and pose $M_t M_t^\top$ by this basis via a small Gram matrix S_t, i.e.,

$$M_t M_t^\top = Q_t S_t Q_t^\top.$$

When A_{t+1} arrives, we decompose it by Q_t and the residual orthogonal to Q_t, as follows

$$A_{t+1} = Q_t Y_{t+1} + R_{t+1}, \qquad R_{t+1} = (I - Q_t Q_t^\top)A_{t+1}.$$

Expanding the basis by the columns of R_{t+1} yields an updated orthonormal matrix Q_{t+1}, and the new Gram matrix S_{t+1} takes a block form constructed from S_t, Y_{t+1}, and the SVD of R_{t+1}.

These results are classical in incremental PCA/SVD, but we provide them for completeness and to make the thesis self-contained.

Lemma 1.11 (Exact incremental Gram factorisation for concatenated blocks)**.** *Let $M_t = [A_1, \dots, A_t] \in \mathbb{R}^{m \times n_t}$ be the horizontal concatenation of*

the first t blocks, and suppose that for some r_t we have an exact factorisation

$$M_t M_t^\top \;=\; Q_t S_t Q_t^\top,$$

where $Q_t \in \mathbb{R}^{m\times r_t}$ has orthonormal columns and $S_t \in \mathbb{R}^{r_t\times r_t}$ is symmetric positive semidefinite. Let a new block $A_{t+1} \in \mathbb{R}^{m\times k}$ be given and define

$$Y := Q_t^\top A_{t+1}, \qquad R := A_{t+1} - Q_t Y.$$

Compute a thin QR decomposition of the residual,

$$R = Q_{\mathrm{res}} B,$$

with $Q_{\mathrm{res}} \in \mathbb{R}^{m\times r_{\mathrm{res}}}$ having orthonormal columns and $B \in \mathbb{R}^{r_{\mathrm{res}}\times k}$, and set

$$Q_{t+1} := \begin{bmatrix} Q_t & Q_{\mathrm{res}} \end{bmatrix} \in \mathbb{R}^{m\times(r_t+r_{\mathrm{res}})}.$$

Define the extended Gram matrix

$$S_{t+1} := \begin{bmatrix} S_t + YY^\top & YB^\top \\ BY^\top & BB^\top \end{bmatrix} \in \mathbb{R}^{(r_t+r_{\mathrm{res}})\times(r_t+r_{\mathrm{res}})}.$$

Then Q_{t+1} has orthonormal columns and

$$M_{t+1} M_{t+1}^\top \;=\; Q_{t+1} S_{t+1} Q_{t+1}^\top,$$

where $M_{t+1} := [M_t, A_{t+1}]$.

Proof. By construction, Q_t has orthonormal columns and Q_{res} is the Q-factor of a thin QR decomposition of the residual R. Moreover, $R = (I - Q_t Q_t^\top)A_{t+1}$ lies in the orthogonal complement of $\mathrm{range}(Q_t)$, hence $Q_t^\top Q_{\mathrm{res}} = 0$, and therefore Q_{t+1} has orthonormal columns.

We first expand the new Gram matrix explicitly:

$$M_{t+1} M_{t+1}^\top = M_t M_t^\top + A_{t+1} A_{t+1}^\top.$$

Using the decomposition $A_{t+1} = Q_t Y + R$ and $M_t M_t^\top = Q_t S_t Q_t^\top$, we get

$$\begin{aligned} M_{t+1} M_{t+1}^\top &= Q_t S_t Q_t^\top + (Q_t Y + R)(Q_t Y + R)^\top \\ &= Q_t S_t Q_t^\top + Q_t Y Y^\top Q_t^\top + Q_t Y R^\top + R Y^\top Q_t^\top + R R^\top. \end{aligned}$$

The orthogonality relation $Q_t^\top R = 0$ implies $R = Q_{\text{res}} B$ for some B, namely the R-factor of the QR decomposition. Substituting this into the above yields

$$\begin{aligned} M_{t+1} M_{t+1}^\top &= Q_t (S_t + Y Y^\top) Q_t^\top + Q_t Y B^\top Q_{\text{res}}^\top \\ &\quad + Q_{\text{res}} B Y^\top Q_t^\top + Q_{\text{res}} B B^\top Q_{\text{res}}^\top \\ &= \begin{bmatrix} Q_t & Q_{\text{res}} \end{bmatrix} \begin{bmatrix} S_t + Y Y^\top & Y B^\top \\ B Y^\top & B B^\top \end{bmatrix} \begin{bmatrix} Q_t & Q_{\text{res}} \end{bmatrix}^\top \\ &= Q_{t+1} S_{t+1} Q_{t+1}^\top, \end{aligned}$$

as claimed. □

Corollary 1.12 (Truncated incremental top-r approximation)**.** *In the setting of Lemma 1.11, let*

$$S_{t+1} = U \Lambda U^\top$$

be an eigendecomposition of S_{t+1} with eigenvalues ordered as $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_{r_t + r_{\text{res}}} \geq 0$. For a target rank $r \leq r_t + r_{\text{res}}$, define

$$U_r := \begin{bmatrix} u_1 & \cdots & u_r \end{bmatrix}, \qquad \Lambda_r := \operatorname{diag}(\lambda_1, \ldots, \lambda_r).$$

Set

$$\widetilde{Q}_{t+1} := Q_{t+1} U_r \in \mathbb{R}^{m \times r}, \qquad \widetilde{S}_{t+1} := \Lambda_r \in \mathbb{R}^{r \times r}.$$

Then:

1. *The matrix $\widetilde{G}_{t+1} := \widetilde{Q}_{t+1} \widetilde{S}_{t+1} \widetilde{Q}_{t+1}^\top$ is the best rank-r approximation to $G_{t+1} := M_{t+1} M_{t+1}^\top$ within the subspace $\operatorname{range}(Q_{t+1})$, in both spectral and Frobenius norms, i.e.*

$$\|G_{t+1} - \widetilde{G}_{t+1}\|_F^2 = \sum_{j > r} \lambda_j(S_{t+1}), \qquad \|G_{t+1} - \widetilde{G}_{t+1}\|_2 = \lambda_{r+1}(S_{t+1}).$$

2. *The top r approximate singular values of M_{t+1} produced by this scheme are*

$$\widetilde{\sigma}_j(M_{t+1}) := \sqrt{\lambda_j(S_{t+1})}, \qquad j = 1, \ldots, r.$$

Proof. By Lemma 1.11, $G_{t+1} = Q_{t+1} S_{t+1} Q_{t+1}^\top$ with Q_{t+1} orthonormal. Any rank-r approximation $\widehat{G}$ whose range is contained in range(Q_{t+1}) can be written as $\widehat{G} = Q_{t+1} X Q_{t+1}^\top$ with $\operatorname{rank}(X) \le r$. Because Q_{t+1} is orthonormal, the Frobenius and spectral norms satisfy

$$\|G_{t+1} - \widehat{G}\| = \|S_{t+1} - X\|,$$

for both $\|\cdot\|_F$ and $\|\cdot\|_2$. By the Eckart–Young–Mirsky theorem (Theorem 1.7), the best rank-r approximation to S_{t+1} in Frobenius and spectral norms is $X = U_r \Lambda_r U_r^\top$, with errors $\sum_{j>r} \lambda_j(S_{t+1})$ and $\lambda_{r+1}(S_{t+1})$, respectively. Substituting $X = U_r \Lambda_r U_r^\top$ and noting that $Q_{t+1} U_r = \widetilde{Q}_{t+1}$ and $\Lambda_r = \widetilde{S}_{t+1}$ gives the first claim.

The second statement is just the observation that the eigenvalues of $\widetilde{G}_{t+1}$ equal $\lambda_1(S_{t+1}), \ldots, \lambda_r(S_{t+1})$, so the corresponding approximate singular values of M_{t+1} are their square roots. □

This identity, stated formally in Lemma 1.11, is exact and well known in the incremental PCA/SVD literature. It provides the algebraic foundation for our blockwise concatenation analysis.

After expanding the basis and updating S_{t+1}, its size grows by the rank of R_{t+1}. To control complexity, we compress back to the rank r by retaining only the top r eigenpairs of S_{t+1}. By Theorem 1.7, this produces the optimal rank-r approximation *within the expanded subspace*. The approximate singular values of M_{t+1} are then given by the square roots of the retained eigenvalues.

The only source of approximation in our incremental estimator is this truncation step, all other steps are exact. A full characterization of the truncation and its implications is provided in Corollary 1.12.

Stability of the incremental estimator.. Although the truncated incremental scheme does not provide deterministic upper or lower bounds on the true truncated SVD error, its approximation quality is governed by classical subspace stability principles. In particular, when the singular value gap

$\sigma_r(M_t) - \sigma_{r+1}(M_t)$ is sufficiently large, truncation preserves the dominant invariant subspace up to small perturbations, and the retained eigenvalues of the compressed Gram matrix remain close to the true leading singular values. This behavior is well documented in the incremental and randomized SVD literature, where approximation error is controlled by the truncation gap and the energy of discarded components.

1.3. Pruning of neural networks

Pruning removes selected weights (sets them to zero) and keeps the rest unchanged. Given parameters Θ, a mask $M \in \{0,1\}^{\dim(\Theta)}$ yields

$$\widehat{\Theta} = M \odot \Theta,$$

so pruning is explicitly a structured perturbation.

Historically influential second-order methods include Optimal Brain Damage and Optimal Brain Surgeon, which use Hessian information to estimate loss increase from removing parameters [71,72]. Modern large-model methods adapt this idea in tractable layerwise forms, including SparseGPT-like and Wanda-like procedures for training-free LLM pruning [7,47]. Recent research also explores attribution-based and movement-based criteria [73–75].

Two major structural choices are common:

- **Unstructured pruning:** remove arbitrary individual weights (high flexibility, irregular sparsity).
- **Structured or semi-structured pruning:** remove weights in hardware-friendly patterns (better acceleration potential, stronger constraints).

For modern accelerators, semi-structured formats such as 2:4 sparsity receive significant attention [76].

1.3.1. Optimal brain damage. We begin by revisiting classical pruning approaches such as Optimal Brain Damage [71], which motivate the rationale behind our approach.

The typical pruning objective is to minimize the error introduced by approximating the original weight matrix. Consider the following objective function:

$$E = \|WX - \widehat{W}X\|_2^2 \to \min, \tag{1.1}$$

where W is the original weight matrix of a layer, $\widehat{W}$ is the pruned (sparse) weight matrix, and X is the input to that layer.

The variation of the error E for a weight row w can be expressed as:

$$\delta E = \left(\frac{\partial E}{\partial w}\right)^\top \delta w + \frac{1}{2}\,\delta w^\top H\,\delta w + \mathcal{O}(\|\delta w\|^3),$$

where $H \equiv \frac{\partial^2 E}{\partial w^2}$ is the Hessian matrix.

At a local minimum of the training error, we have

$$\frac{\partial E}{\partial w} \approx 0,$$

and higher order terms are neglected.

Our goal is to set one of the weights, say w_q, to zero while minimizing the increase in error. This introduces the constraint:

$$e_q^\top \delta w + w_q = 0,$$

where e_q is the qth standard basis vector. Thus, the optimization problem in equation (1.1) can be reformulated as:

$$\min_{\delta w} \quad \frac{1}{2}\,\delta w^\top H\,\delta w, \qquad \text{s.t.} \qquad e_q^\top \delta w + w_q = 0. \tag{1.2}$$

This constrained problem can be solved using Lagrange multipliers.

$$\min_{\delta w} \quad \frac{1}{2}\delta w^\top H\,\delta w, \qquad \text{s.t.} \qquad e_q^\top \delta w + w_q = 0.$$

The problem could be solved using Lagrange multiplier. We begin with the Lagrangian:

$$\mathcal{L} = \frac{1}{2}\delta w^\top H \delta w + \lambda \left(e_q^\top \delta w + w_q \right).$$

Taking the derivative with respect to δw and setting it to zero:

$$\nabla_{\delta w}\mathcal{L} = H\delta w + \lambda e_q = 0 \quad \Rightarrow \quad \delta w = -H^{-1}e_q\lambda.$$

Substituting into the constraint:

$$e_q^\top(-H^{-1}e_q\lambda) + w_q = 0,$$

we get:

$$\lambda = \frac{w_q}{e_q^\top H^{-1} e_q}.$$

Thus, the change in weights:

$$\delta w = -H^{-1}e_q \cdot \frac{w_q}{e_q^\top H^{-1} e_q}.$$

Notice that:

$$H\delta w = H\left(-H^{-1}e_q \frac{w_q}{e_q^\top H^{-1} e_q}\right) = -e_q \frac{w_q}{e_q^\top H^{-1} e_q},$$

and

$$\delta w^\top = \left(-H^{-1}e_q \frac{w_q}{e_q^\top H^{-1} e_q}\right)^\top = -\frac{w_q}{e_q^\top H^{-1} e_q} e_q^\top H^{-1}$$

Now compute the increase in error:

$$E_q = \frac{1}{2}\delta w^\top H \delta w = \frac{1}{2}\frac{w_q}{e_q^\top H^{-1} e_q} e_q^\top H^{-1} e_q \frac{w_q}{e_q^\top H^{-1} e_q} = \frac{1}{2}\cdot\frac{w_q^2}{e_q^\top H^{-1} e_q}$$

The resulting increase in error is given by:

$$E_q = \frac{1}{2}\cdot\frac{w_q^2}{e_q^\top H^{-1} e_q}. \tag{1.3}$$

By computing E_q for every weight w_q, one can prune the weight that causes the smallest increase in error, thereby minimally affecting the layer's output. Intuitively, this means we identify which weights are most critical for the model's performance on a specific language. Weights with high importance scores are those whose removal would substantially degrade the model's ability to predict tokens in that language.

Chapter 2

Singular values perturbations and low-rank approximations of concatenated matrices

The problem of compressing large datasets arises in diverse applications [77–80], where data are often organized as a *collection* of related matrices rather than a single monolithic array. Typical examples include time windows, sensor modalities, neural-network layers, federated clients, and task-specific blocks.

This chapter explicitly studies three coupled objects: *concatenated matrices*, *perturbations of their singular values*, and *compression-aware clustering*. The core design question is: *given N matrices, when should we compress them separately, and when should we concatenate them and apply a joint low-rank approximation?*

Given matrices $\{A_i\}_{i=1}^{N}$ with a common row dimension, we consider their horizontal concatenation

$$M = [A_1\ A_2\ \cdots\ A_N].$$

Joint compression of M can exploit a shared low-dimensional basis across blocks and increase compression efficiency. However, concatenation also changes the spectrum of the resulting matrix, so a shared approximation may become inaccurate if block subspaces are weakly aligned.

To analyze this trade-off, we study blockwise perturbations $\widetilde{A}_i = A_i + E_i$ and the induced concatenated matrix

$$\widetilde{M} = [\widetilde{A}_1, \widetilde{A}_2, \ldots, \widetilde{A}_k] = M + E, \qquad M = [A_1, A_2, \ldots, A_k] \in \mathbb{R}^{m\times kn}.$$

The central spectral question is how the dominant singular values of $\widetilde{M}$ deviate from those of M as functions of the block perturbations $\{E_i\}$ and the group size k.

The perturbation analysis developed here characterizes spectral sensitivity under blockwise concatenation and provides theoretical insight into when joint compression may become unstable. The *compression-aware clustering* framework, however, is formulated through explicit reconstruction-feasibility constraints: groups are accepted only when their joint low-rank approximation satisfies a prescribed error budget. In contrast to heuristic grouping rules [56,57,59], this chapter develops quantitative criteria for deciding when concatenation is practically beneficial.

2.1. Singular value perturbations of concatenated matrices

In this section, we establish an upper bound on the perturbation error for a concatenated matrix.

Let $\{A_i\}_{i=1}^k$ be a collection of matrices with $A_i \in \mathbb{R}^{m\times n}$ for each $i = 1,\dots,k$. Define the original concatenated matrix and its perturbed one as

$$M = [A_1, A_2, \dots, A_k] \quad \text{and} \quad \widetilde{M} = [\widetilde{A}_1, \widetilde{A}_2, \dots, \widetilde{A}_k],$$

respectively, so that the perturbation is

$$E = \widetilde{M} - M = [E_1, E_2, \dots, E_k],$$

where $E_i = \widetilde{A}_i - A_i$ for $i = 1,\dots,k$.

Classical perturbation bounds are based on Weyl's inequality [53] and its revisions by Davis–Kahan and Stewart–Sun [52, 81, 82]. They estimate the change in singular values of *single matrix* under global perturbations and reveal neither (i) how blockwise errors accumulate inside a concatenation, nor (ii) whether working with $M^\top M$ or $MM^\top$ leads to a sharper bound in practice.

Consider now a matrix formed by the concatenation of the same copy of matrix A. The structure of the constructed matrix induces a certain regularity that we can exploit.

Proposition 2.1. *Let $A \in \mathbb{R}^{m\times n}$ be a matrix with the* $\operatorname{rank}(A) = r$. *Define the concatenated matrix*

$$M \;=\; [\, A,\, A,\, \ldots,\, A\,] \in \mathbb{R}^{m\times kn},$$

which consists of k copies of A. Define

$$S_1 = M^\top M \quad \textit{and} \quad S_2 = MM^\top.$$

Then:

- *The nonzero eigenvalues of both S_1 and S_2 are as follows*

$$\lambda_i(S_1) = \lambda_i(S_2) = k\,\lambda_i(A^\top A), \quad 1 \le i \le r.$$

- *The remaining eigenvalues are the zero, i.e.,*

$$\lambda_i(S_1) = 0 \quad \textit{for } r < i \le kn,$$

$$\lambda_i(S_2) = 0 \quad \textit{for } r < i \le m.$$

Proof. Note that

$$S_2 = MM^\top = [\, A,\, A,\, \ldots,\, A\,] \begin{bmatrix} A^\top \\ A^\top \\ \vdots \\ A^\top \end{bmatrix} = k\,AA^\top.$$

If x_i is an eigenvector of $A^\top A$ corresponding to the eigenvalue $\lambda_i(A^\top A)$, then due to properties of the SVD, the nonzero eigenvalues of $AA^\top$ coincide with those of $A^\top A$. Consequently,

$$S_2\, x_i = k\, AA^\top x_i = k\,\lambda_i(A^\top A)\, x_i, \quad i = 1, \ldots, r.$$

Therefore,

$$\lambda_i(S_2) = k\,\lambda_i(A^\top A), \quad i = 1, \ldots, r,$$

with the remaining eigenvalues being zero. Since $M^\top M$ and $MM^\top$ share the same nonzero eigenvalues, the claim for S_1 follows. □

The following proposition establishes a bound for a block matrix. This bound will be applied when analyzing block-structured perturbations.

Proposition 2.2 (Bounding the norm of a block matrix)**.** *Let*

$$E = \begin{bmatrix} E_{11} & E_{12} & \ldots & E_{1k} \\ E_{21} & E_{22} & \ldots & E_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ E_{k1} & E_{k2} & \ldots & E_{kk} \end{bmatrix}$$

be a block matrix. Then,

$$\|E\|_2 \leq \sqrt{\sum_{i=1}^{k}\sum_{j=1}^{k} \|E_{ij}\|_2^2}.$$

Proof. For any nonzero vector

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_k \end{bmatrix},$$

we have

$$Ex = \begin{bmatrix} \sum_{j=1}^{k} E_{1j}x_j \\ \sum_{j=1}^{k} E_{2j}x_j \\ \vdots \\ \sum_{j=1}^{k} E_{kj}x_j \end{bmatrix}.$$

Define $y_i = \sum_{j=1}^{k} E_{ij}x_j$ for $i = 1, \ldots, k$. The triangle inequality and sub-multiplicative property of the spectral norm deduce

$$\|y_i\|_2 \leq \sum_{j=1}^{k} \|E_{ij}x_j\|_2 \leq \sum_{j=1}^{k} \|E_{ij}\|_2\|x_j\|_2$$

and, according to the Cauchy–Schwarz inequality,

$$\|y_i\|_2^2 \leq \left(\sum_{j=1}^{k} \|E_{ij}\|_2^2\right)\left(\sum_{j=1}^{k} \|x_j\|_2^2\right).$$

Because $\|x\|_2^2 = \sum_{j=1}^{k} \|x_j\|_2^2$, summation over i yields

$$\|Ex\|_2^2 = \sum_{i=1}^{k} \|y_i\|_2^2 \leq \left(\sum_{i=1}^{k}\sum_{j=1}^{k} \|E_{ij}\|_2^2\right) \|x\|_2^2,$$

so that taking the square root and subsequently maximizing over all $x \neq 0$ gives

$$\|E\|_2 \leq \sqrt{\sum_{i=1}^{k}\sum_{j=1}^{k} \|E_{ij}\|_2^2}. \qquad \square$$

The following lemmas characterize the perturbation structure for the two Gram matrices $M^\top M$ and $MM^\top$.

Lemma 2.3. *The perturbation term $D = \widetilde{M}^\top \widetilde{M} - M^\top M$ has the upper bound*

$$\|D\|_2 \leq \sqrt{\sum_{i=1}^{k}\sum_{j=1}^{k} \left(\|A_i\|_2\|E_j\|_2 + \|E_i\|_2\|A_j\|_2 + \|E_i\|_2\|E_j\|_2\right)^2}.$$

Proof. Recalling $\widetilde{A}_i = A_i + E_i$ implies

$$\widetilde{M}^\top \widetilde{M} = \begin{bmatrix} (A_1 + E_1)^\top \\ \vdots \\ (A_k + E_k)^\top \end{bmatrix} \left[(A_1 + E_1), \ldots, (A_k + E_k)\right]$$

so that expanding the product yields

$$\widetilde{M}^\top \widetilde{M} = \begin{bmatrix} (A_1 + E_1)^\top (A_1 + E_1) & \ldots & (A_1 + E_1)^\top (A_k + E_k) \\ \vdots & \ddots & \vdots \\ (A_k + E_k)^\top (A_1 + E_1) & \ldots & (A_k + E_k)^\top (A_k + E_k) \end{bmatrix},$$

where each block takes the form

$$(A_i + E_i)^\top (A_j + E_j) = A_i^\top A_j + A_i^\top E_j + E_i^\top A_j + E_i^\top E_j.$$

Therefore, the unperturbed term takes the form

$$M^\top M = \begin{bmatrix} A_1^\top A_1 & \dots & A_1^\top A_k \\ \vdots & \ddots & \vdots \\ A_k^\top A_1 & \dots & A_k^\top A_k \end{bmatrix},$$

and the perturbation matrix D reads as

$$\begin{aligned} D &= \begin{bmatrix} A_1^\top E_1 + E_1^\top A_1 + E_1^\top E_1 & \dots & A_1^\top E_k + E_1^\top A_k + E_1^\top E_k \\ \vdots & \ddots & \vdots \\ A_k^\top E_1 + E_k^\top A_1 + E_k^\top E_1 & \dots & A_k^\top E_k + E_k^\top A_k + E_k^\top E_k \end{bmatrix} \\ &= \begin{bmatrix} D_{11} & \dots & D_{1k} \\ \vdots & \ddots & \vdots \\ D_{k1} & \dots & D_{kk} \end{bmatrix}. \end{aligned}$$

where the individual block components are $D_{ij} = A_i^\top E_j + E_i^\top A_j + E_i^\top E_j$.

By applying the triangle inequality and the submultiplicative property of the norm,

$$\begin{aligned} \|D_{ij}\|_2 &\le \|A_i^\top E_j\|_2 + \|E_i^\top A_j\|_2 + \|E_i^\top E_j\|_2 \\ &\le \|A_i\|_2 \|E_j\|_2 + \|E_i\|_2 \|A_j\|_2 + \|E_i\|_2 \|E_j\|_2. \end{aligned}$$

By Proposition 2.2,

$$\|D\|_2 \le \sqrt{\sum_{i=1}^{k} \sum_{j=1}^{k} \|D_{ij}\|_2^2},$$

which yields

$$\|D\|_2 \le \sqrt{\sum_{i=1}^{k} \sum_{j=1}^{k} \left(\|A_i\|_2 \|E_j\|_2 + \|E_i\|_2 \|A_j\|_2 + \|E_i\|_2 \|E_j\|_2 \right)^2}.$$

□

Lemma 2.4. *The perturbation term* $D = \widetilde{M}\widetilde{M}^\top - MM^\top$ *has the upper bound*

$$\|D\|_2 \leq \sum_{i=1}^{k} \left(2\|A_i\|_2\|E_i\|_2 + \|E_i\|_2^2\right).$$

Proof. We begin by expressing $\widetilde{M}\widetilde{M}^\top$ using the concatenated form of $\widetilde{M}$, i.e.,

$$\widetilde{M}\widetilde{M}^\top = [\widetilde{A}_1, \ldots, \widetilde{A}_k] \begin{bmatrix} \widetilde{A}_1^\top \\ \vdots \\ \widetilde{A}_k^\top \end{bmatrix}$$

which leads to the sum

$$\widetilde{M}\widetilde{M}^\top = \sum_{i=1}^{k} \widetilde{A}_i\widetilde{A}_i^\top.$$

Next, recall that $\widetilde{A}_i = A_i + E_i$; hence,

$$\widetilde{A}_i\widetilde{A}_i^\top = (A_i + E_i)(A_i + E_i)^\top = A_iA_i^\top + A_iE_i^\top + E_iA_i^\top + E_iE_i^\top.$$

As a consequence, we can write the sum as

$$\widetilde{M}\widetilde{M}^\top = \sum_{i=1}^{k} \left(A_iA_i^\top + A_iE_i^\top + E_iA_i^\top + E_iE_i^\top\right).$$

The original matrix $MM^\top = \sum_{i=1}^{k} A_iA_i^\top$, therefore, the perturbation term

$$D = \widetilde{M}\widetilde{M}^\top - MM^\top = \sum_{i=1}^{k} \left(A_iE_i^\top + E_iA_i^\top + E_iE_i^\top\right).$$

To estimate the bound $\|D\|_2$, we consequently apply the triangle inequality and the submultiplicative property of the spectral norm:

$$\|D\|_2 \leq \sum_{i=1}^{k}(\|A_iE_i^\top\|_2 + \|E_iA_i^\top\|_2 + \|E_iE_i^\top\|_2) \leq \sum_{i=1}^{k} \left(2\|A_i\|_2\|E_i\|_2 + \|E_i\|_2^2\right).$$

□

Lemmas 2.3 and 2.4 provide explicit bounds for the perturbations of the two Gram matrices $M^\top M$ and $MM^\top$. Notably, the perturbation of $MM^\top$ depends only on *within-block* terms, whereas $M^\top M$ couples *all* pairs of blocks (i, j). This structural difference yields a tighter and more interpretable estimate when working with $MM^\top$, and we will exploit it in the main perturbation bound below.

Theorem 2.5 (Singular value perturbation of concatenated matrix)**.** *Let $\{A_i\}_{i=1}^k$ be a collection of matrices with $A_i \in \mathbb{R}^{m\times n}$ for all $i = 1, \dots, k$. Define the original concatenated matrix by*

$$M = [A_1, \dots, A_k] \in \mathbb{R}^{m\times kn}, \quad r = \operatorname{rank}(M).$$

Also, let $\widetilde{M} = [\widetilde{A}_1, \dots, \widetilde{A}_k]$ be a perturbed version of M, and define the perturbation matrix

$$E = \widetilde{M} - M = [E_1, \dots, E_k], \quad \text{with } E_i = \widetilde{A}_i - A_i \text{ for each } i = 1, \dots, k.$$

Then, the following perturbation bounds for the singular values hold true:

- *For $i = 1, \dots, r$ (corresponding to the nonzero singular values of M),*

$$|\sigma_i(\widetilde{M}) - \sigma_i(M)| \le \frac{1}{\sigma_i(M)} \sum_{j=1}^{k} \Big(2\|A_j\|_2\|E_j\|_2 + \|E_j\|_2^2\Big).$$

- *For $i = r+1, \dots, \min(m, kn)$ (corresponding to the zero singular values of M),*

$$\sigma_i(\widetilde{M}) \le \sqrt{\sum_{j=1}^{k} \Big(2\|A_j\|_2\|E_j\|_2 + \|E_j\|_2^2\Big)}.$$

Proof. We start with Weyl's inequality applied to the two corresponding symmetric matrices $M^\top M$ and $MM^\top$, to derive bounds on perturbations of their eigenvalues, which are linked to the singular values of M and $\widetilde{M}$.

Consider $M^\top M$ and using the Lemma 2.3, write down

$$\|D_1\|_2 \le \sqrt{\sum_{i=1}^{k}\sum_{j=1}^{k}\Big(\|A_i\|_2\|E_j\|_2 + \|E_i\|_2\|A_j\|_2 + \|E_i\|_2\|E_j\|_2\Big)^2},$$

where the perturbation term $D_1 = \widetilde{M}^\top\widetilde{M} - M^\top M$.

Applying Weyl's inequality to the eigenvalues of $M^\top M$ and $\widetilde{M}^\top\widetilde{M}$ leads to

$$|\lambda_i(\widetilde{M}^\top\widetilde{M}) - \lambda_i(M^\top M)| \le \|D_1\|_2.$$

According to Proposition 1.8, the eigenvalues of $M^\top M$ and $\widetilde{M}^\top\widetilde{M}$ are squares of the singular values, i.e.,

$$\lambda_i(\widetilde{M}^\top\widetilde{M}) = \sigma_i^2(\widetilde{M}) \quad \text{and} \quad \lambda_i(M^\top M) = \sigma_i^2(M) \;\Rightarrow\; |\sigma_i^2(\widetilde{M}) - \sigma_i^2(M)| \le \|D_1\|_2.$$

In similar way, according to Lemma 2.4

$$\|D_2\|_2 \le \sum_{i=1}^{k}\Big(2\|A_i\|_2\|E_i\|_2 + \|E_i\|_2^2\Big),$$

where $D_2 = \widetilde{M}\widetilde{M}^\top - MM^\top$. Weyl's inequality implies

$$|\lambda_i(\widetilde{M}\widetilde{M}^\top) - \lambda_i(MM^\top)| \le \|D_2\|_2,$$

and, again, due to Proposition 1.8

$$\lambda_i(\widetilde{M}\widetilde{M}^\top) = \sigma_i^2(\widetilde{M}) \quad \text{and} \quad \lambda_i(MM^\top) = \sigma_i^2(M) \;\Rightarrow\; |\sigma_i^2(\widetilde{M}) - \sigma_i^2(M)| \le \|D_2\|_2.$$

Because bounds provided by Lemma 2.4 (and hence $\|D_2\|_2$) are stronger than those of Lemma 2.3, we continue the proof with $\|D_2\|_2$.

For $\sigma_i(M) > 0$, $i = 1, \dots, r$, note that

$$\sigma_i^2(\widetilde{M}) - \sigma_i^2(M) = \big(\sigma_i(\widetilde{M}) - \sigma_i(M)\big)\big(\sigma_i(\widetilde{M}) + \sigma_i(M)\big).$$

Since singular values are nonnegative and $\sigma_i(\widetilde{M}) + \sigma_i(M) > 0$,

$$|\sigma_i(\widetilde{M}) - \sigma_i(M)| \le \frac{\|D_2\|_2}{\sigma_i(\widetilde{M}) + \sigma_i(M)}.$$

Moreover, because $\sigma_i(\widetilde{M}) + \sigma_i(M) > \sigma_i(M)$,

$$|\sigma_i(\widetilde{M}) - \sigma_i(M)| \le \frac{\|D_2\|_2}{\sigma_i(M)} \le \frac{1}{\sigma_i(M)} \sum_{j=1}^{k} \Big(2\|A_j\|_2 \|E_j\|_2 + \|E_j\|_2^2 \Big).$$

For indices $i = r+1, \dots, \min(m, kn)$, $\sigma_i(M) = 0$. In this case,

$$\sigma_i^2(\widetilde{M}) \le \|D_2\|_2,$$

which implies

$$\sigma_i(\widetilde{M}) \le \sqrt{\|D_2\|_2} \le \sqrt{\sum_{j=1}^{k} \Big(2\|A_j\|_2 \|E_j\|_2 + \|E_j\|_2^2 \Big)}.$$

□

Two useful corollaries of Theorem 2.5 arise in a "centroid" setting, where the concatenation is formed by repeating a single reference block A and perturbing only some of the copies.

Corollary 2.6 (Perturbation around the centroid)**.** *Let $A \in \mathbb{R}^{m\times n}$ with rank $r = \operatorname{rank}(A)$. Consider perturbations $\{E_i\}_{i=2}^{k}$ with $E_i \in \mathbb{R}^{m\times n}$ for $i = 2, \dots, k$ and define the concatenated matrix by*

$$M = [A, A, \dots, A] \in \mathbb{R}^{m\times kn},$$

whereas the perturbed matrix is

$$\widetilde{M} = [A, A + E_2, A + E_3, \dots, A + E_k] \in \mathbb{R}^{m\times kn}$$

(here $E_1 \equiv 0$).

Then,

$$|\sigma_i(\widetilde{M}) - \sqrt{k}\,\sigma_i(A)| \le \frac{1}{\sqrt{k}\,\sigma_i(A)} \sum_{j=2}^{k} \Big(2\|A\|_2 \|E_j\|_2 + \|E_j\|_2^2 \Big), \quad i = 1, \dots, r,$$

and

$$\sigma_i(\widetilde{M}) \le \sqrt{\sum_{j=2}^{k} \Big(2\|A\|_2 \|E_j\|_2 + \|E_j\|_2^2 \Big)}, \quad i = r+1, \dots, \min(m, kn).$$

Proof. The result follows from Theorem 2.5. Because $A_j = A$ for all j and $E_1 \equiv 0$, the bound in Theorem 2.5 becomes

$$|\sigma_i(\widetilde{M}) - \sigma_i(M)| \leq \frac{1}{\sigma_i(M)} \sum_{j=2}^{k} \Big(2\|A\|_2 \|E_j\|_2 + \|E_j\|_2^2 \Big).$$

Furthermore, from Propositions 1.8 and 2.1 it follows that

$$\sigma_i^2(M) = \lambda_i(M^\top M) = k\,\lambda_i(A^\top A) = k\,\sigma_i^2(A) \;\Rightarrow\; \sigma_i(M) = \sqrt{k}\,\sigma_i(A).$$

Substituting this into the previous inequality yields the result. □

An immediate consequence of Corollary 2.6 is a continuity statement: if each perturbation is small, then the singular values of the perturbed concatenation converge to those of the unperturbed (scaled) matrix.

Corollary 2.7 (Continuity of Singular Values under Small Perturbations)**.** *Suppose that* $\|E_j\|_2 < \epsilon$ *for all* j*. Then,*

$$\lim_{\epsilon \to 0} \sigma_i(\widetilde{M}) = \sqrt{k}\,\sigma_i(A) \quad i = 1, \ldots, r,$$

and

$$\lim_{\epsilon \to 0} \sigma_i(\widetilde{M}) = 0 \quad i = r+1, \ldots, \min(m, kn).$$

Proof. Corollary 2.6 leads to

$$\begin{aligned} |\sigma_i(\widetilde{M}) - \sqrt{k}\,\sigma_i(A)| &\leq \frac{1}{\sqrt{k}\,\sigma_i(A)} \sum_{j=2}^{k} \Big(2\|A\|_2 \|E_j\|_2 + \|E_j\|_2^2 \Big) \\ &\leq \frac{1}{\sqrt{k}\,\sigma_i(A)} \sum_{j=2}^{k} \Big(2\|A\|_2 \epsilon + \epsilon^2 \Big) \\ &= \frac{(k-1)\epsilon}{\sqrt{k}\,\sigma_i(A)} \Big(2\|A\|_2 + \epsilon \Big), \quad i = 1, \ldots, r. \end{aligned}$$

Therefore, it follows that

$$\lim_{\epsilon \to 0} |\sigma_i(\widetilde{M}) - \sqrt{k}\,\sigma_i(A)| = 0.$$

For $i = r+1, \ldots, \min(m, kn)$, Corollary 2.6 implies

$$\sigma_i(\widetilde{M}) \le \sqrt{\sum_{j=2}^{k} \Big(2\|A\|_2\|E_j\|_2 + \|E_j\|_2^2\Big)} \le \sqrt{\sum_{j=2}^{k} \Big(2\|A\|_2\epsilon + \epsilon^2\Big)}$$
$$= \sqrt{(k-1)\epsilon\Big(2\|A\|_2 + \epsilon\Big)} \;\Rightarrow\; \lim_{\epsilon \to 0} \sigma_i(\widetilde{M}) = 0. \qquad \square$$

Modern sensing and communication systems often produce large collections of matrix-valued observations that share an approximately common column space. A natural way to exploit this redundancy is to *concatenate* multiple blocks and store a *single* joint rank-r singular value decomposition (SVD), rather than computing separate truncated SVDs for each block. This yields a tradeoff:

> **Storage efficiency.** Concatenating more blocks reduces the number of scalars required per stored singular vector.
>
> **Spectral perturbation.** However, concatenating too many noisy blocks perturbs the leading singular values and singular vectors, potentially degrading downstream tasks that rely on them.

The practical task is to design a *computationally efficient rule* that determines *in advance* how many blocks can be safely merged without exceeding a prescribed absolute tolerance τ for the top singular values. This is addressed by the next corollary.

Definition 2.8 (Spectral budget)**.** Fix a target rank $r \in \mathbb{N}$ and an absolute tolerance $\tau > 0$. We call the pair (r, τ) the *spectral budget* and say that a perturbed matrix $\widetilde{M}$ satisfies the (r, τ)-*spectral budget* with respect to a reference matrix M if

$$\big|\sigma_i(\widetilde{M}) - \sigma_i(M)\big| \;\le\; \tau, \qquad i = 1, \ldots, r.$$

Corollary 2.9 (Maximum concatenation length under a (r,τ)-spectral budget)**.** *Let $A_0 \in \mathbb{R}^{m\times n}$ be fixed and consider*

$$M_{1:k} = \big[A_0, \dots, A_0\big] \in \mathbb{R}^{m\times kn},$$
$$\widetilde{M}_{1:k} = \big[A_0,\ A_0 + E_2, \dots, A_0 + E_k\big] \in \mathbb{R}^{m\times kn},$$
$$k \geq 1.$$

Fix a (r,τ)-spectral budget, which should hold

$$\big|\sigma_i(\widetilde{M}_{1:k}) - \sigma_i(M_{1:k})\big| \leq \tau, \quad i = 1, \dots, r. \tag{2.1}$$

Then the largest group size $k_{\max}$ for which (2.1) can be guaranteed is

$$k \leq k_{\max}(\tau) := \left(\frac{\tau\,\sigma_r(A_0)}{2\|A_0\|_2\,\bar{\varepsilon}(k) + \bar{\varepsilon}(k)^2}\right)^2, \tag{2.2}$$

where $\bar{\varepsilon}(k) := \max_{2\leq j\leq k} \|E_j\|_2$.

Proof. Because $M_{1:k}$ repeats A_0 horizontally, from Proposition 2.1

$$\sigma_i(M_{1:k}) = \sqrt{k}\,\sigma_i(A_0), \quad i = 1, \dots, r.$$

According to Corollary 2.6,

$$\big|\sigma_i(\widetilde{M}_{1:k}) - \sqrt{k}\,\sigma_i(A_0)\big| \ \leq\ \frac{1}{\sqrt{k}\,\sigma_i(A_0)} \sum_{j=2}^{k} \big(2\|A_0\|_2\|E_j\|_2 + \|E_j\|_2^2\big), \quad i = 1, \dots, r.$$

The inequality $\|E_j\|_2 \leq \bar{\varepsilon}(k)$ implies

$$\sum_{j=2}^{k} \big(2\|A_0\|_2\|E_j\|_2 + \|E_j\|_2^2\big) \leq (k-1)\big(2\|A_0\|_2\bar{\varepsilon}(k) + \bar{\varepsilon}(k)^2\big) \leq k\big(2\|A_0\|_2\bar{\varepsilon}(k) + \bar{\varepsilon}(k)^2\big).$$

Hence, because $\sigma_i(A_0) \geq \sigma_r(A_0)$,

$$\big|\sigma_i(\widetilde{M}_{1:k}) - \sqrt{k}\,\sigma_i(A_0)\big| \leq \frac{1}{\sqrt{k}\,\sigma_i(A_0)} \sum_{j=2}^{k} \big(2\|A_0\|_2\|E_j\|_2 + \|E_j\|_2^2\big)$$

$$\leq \frac{\sqrt{k}\big(2\|A_0\|_2\bar{\varepsilon}(k) + \bar{\varepsilon}(k)^2\big)}{\sigma_r(A_0)}, \quad i = 1, \ldots, r.$$

Requiring the right-hand side to be $\leq \tau$ yields

$$\sqrt{k}\big(2\|A_0\|_2\bar{\varepsilon}(k) + \bar{\varepsilon}(k)^2\big) \leq \tau\, \sigma_r(A_0).$$

Squaring both sides and rearranging gives the claimed condition $k \leq k_{\max}(\tau)$ in (2.2). Monotonicity of $\bar{\varepsilon}(k)$ implies monotonicity of $k_{\max}(\tau)$, which completes the proof. □

Because $\bar{\varepsilon}(k)$ is nondecreasing in k, the right-hand side of (2.2) decreases as blocks are appended. Therefore, the first violation of (2.2) yields the maximal group size that is *guaranteed* to keep every leading singular value within the tolerance τ in (2.1). The resulting criterion provides a simple a priori rule for deciding how many channels to merge *before* performing a large-scale SVD.

2.2. Compression-aware clustering

Concatenated SVD has been successfully applied across a broad range of domains. In large language models, joint SVD of concatenated weight matrices has been used to share low-rank projections across attention components, layers, or experts, enabling parameter reduction while preserving accuracy. Representative examples include unified QKV decompositions [56], intra-layer shared projections [57], cross-layer parameter sharing [58], and expert merging in mixture-of-experts architectures [59, 60]. In these settings, concatenation is typically guided by architectural structure (e.g., matrices belonging to the same layer or module) or semantic similarity (e.g., adjacent layers or related experts). Related ideas also appear in wireless signal processing, where concatenated SVD is used to design shared precoders across frequency bands [61], as well as in neuroscience and genomics, where large collections of measurements are concatenated to obtain global low-dimensional representations across sessions, experimental conditions, or chromosomes [62, 63].

Across these application areas, concatenated SVD serves as a powerful tool for extracting shared structure from collections of matrices.

Despite its empirical success, existing uses of concatenated SVD rely on *predefined or heuristic grouping* of matrices. The decision of which matrices should share a low-rank basis is typically made manually based on domain knowledge, architectural constraints, or simple similarity measures. Crucially, these approaches do not provide a principled mechanism for determining *which subsets of matrices should be concatenated* in order to satisfy a prescribed reconstruction error or compression target. As a result, the quality of compression depends heavily on ad hoc design choices, and there are no guarantees that merging additional matrices will not violate reconstruction constraints. In contrast, the present work formulates matrix grouping as a *compression-driven clustering problem*. Rather than clustering matrices based on ambient-space distances or semantic heuristics, cluster formation is governed directly by predicted low-rank approximation error of the concatenated matrix. This enables principled selection of matrix groups under explicit reconstruction error budgets.

Consider a collection of real matrices

$$\mathcal{A} = \{A_i\}_{i=1}^{N}, \qquad A_i \in \mathbb{R}^{m \times n_i}, \tag{2.3}$$

that must be stored and manipulated under limited memory. Our goal is to replace the original matrices by compressed surrogates $\{\widehat{A}_i\}_{i=1}^{N}$ that preserve essential structure while using as few real numbers as possible.

We measure fidelity using the Frobenius norm

$$\mathcal{L}(\widehat{\mathcal{A}}, \mathcal{A}) = \left(\sum_{i=1}^{N} \|A_i - \widehat{A}_i(\Theta)\|_F^2 \right)^{1/2},$$

where a collection of real values Θ parameterizes the compressed representation. Let $\mathrm{mem}(\Theta)$ denote the number of real values required to store Θ.

The *error-constrained memory minimization* principle reads as

$$\min_{\Theta} \ \mathrm{mem}(\Theta) \quad \text{s.t.} \quad \mathcal{L}(\widehat{\mathcal{A}}, \mathcal{A}) \ \leq \ \varepsilon, \tag{2.4}$$

where $\varepsilon > 0$ prescribes the maximum admissible distortion.

Compressing each matrix independently ignores potential *redundancy across the collection*. We assume that many matrices have similar column spaces, which enables a joint representation via a shared low-rank basis. This motivates clustering followed by a joint factorization.

Recall that $A_i \in \mathbb{R}^{m \times n_i}$ for $i = 1, \ldots, N$, and let $\Pi = \{C_1, \ldots, C_K\}$ be a partition of this index set. For each cluster $C_c \in \Pi$, consider the concatenated matrix

$$M_c = [A_i]_{i \in C_c} \in \mathbb{R}^{m \times N_c}, \qquad \text{where} \quad N_c = \sum_{i \in C_c} n_i.$$

For each cluster C_c, let r_c denote the target rank. We compute a rank-r_c truncated SVD of M_c,

$$M_c \approx U_c S_c V_c^{\top},$$

where $U_c \in \mathbb{R}^{m \times r_c}$ and $V_c \in \mathbb{R}^{N_c \times r_c}$ have orthonormal columns, and $S_c \in \mathbb{R}^{r_c \times r_c}$ contains the leading singular values. This provides a low-rank representation of the concatenated matrix M_c. Since the diagonal matrix S_c can be absorbed into either factor of the decomposition, we keep only two matrices per cluster,

$$\widetilde{U}_c = U_c S_c \in \mathbb{R}^{m \times r_c}, \qquad V_c \in \mathbb{R}^{N_c \times r_c}.$$

This reduces storage while preserving a rank-r_c representation. The memory footprint for cluster C_c is therefore

$$\mathrm{mem}_c = r_c(m + N_c),$$

consisting of mr_c entries for $\widetilde{U}_c$ and $N_c r_c$ for V_c.

To reconstruct an approximation of each original matrix A_i, we split V_c column-wise according to the block dimensions n_i. If $V_{c,i} \in \mathbb{R}^{n_i \times r_c}$ denotes the submatrix corresponding to A_i, then the recovered approximation is given by

$$\widehat{A}_i = \widetilde{U}_c V_{c,i}^\top, \qquad i \in C_c.$$

Thus each A_i is represented using only the shared left factor $\widetilde{U}_c$ and its cluster-specific coefficient block $V_{c,i}$.

Because the rank-r_c truncated SVD is the optimal Frobenius-norm approximation of M_c by the Eckart–Young–Mirsky theorem, the approximation error for cluster C_c is precisely the energy in the discarded singular values:

$$\mathcal{L}_c = \left(\sum_{i \in C_c} \|A_i - \widehat{A}_i\|_F^2 \right)^{1/2} = \left(\sum_{j > r_c} \sigma_j^2(M_c) \right)^{1/2},$$

where $\sigma_j(M_c)$ denotes the j-th largest singular value of M_c, with singular values ordered non-increasingly.

The error-constrained memory minimization problem under cluster–concatenate–SVD representation with two stored matrices per cluster reads as

$$\min_{\Pi, \{r_c\}} \quad \sum_{c=1}^{K} r_c \, (m + N_c) \qquad \text{s.t.} \quad \left(\sum_{c=1}^{K} \sum_{j > r_c} \sigma_j^2(M_c) \right)^{1/2} \leq \varepsilon, \tag{2.5}$$

where $\Pi = \{C_1, \ldots, C_K\}$ is a partition of $\{1, \ldots, N\}$, r_c is the rank assigned to cluster C_c, m is the row dimension of all matrices, N_c is the total number of columns in cluster C_c, and $\varepsilon > 0$ is the user-specified reconstruction tolerance controlling the achievable trade-off between accuracy and compression.

This formulation explicitly balances *storage per cluster* and *approximation error*. Adjusting ε or the cluster-wise ranks r_c tunes the trade-off between memory footprint and approximation quality.

Why not optimize (2.5) *directly?* While the formulation in (2.5) is compact and conceptually appealing, it is important to clarify why our approach

does *not* attempt to solve it directly. The difficulty is twofold. First, the optimization over the partition Π is inherently *combinatorial*. Even for fixed ranks $\{r_c\}$, determining an optimal clustering that minimizes the aggregate spectral tail $\sum_c \sum_{j>r_c} \sigma_j^2(M_c)$ subsumes hard partitioning problems and is NP-hard in general. Thus, global optimization over Π is computationally intractable beyond very small instances. Second, even evaluating the objective is expensive. For a candidate cluster C_c, computing $\sigma_j(M_c)$ requires forming the concatenated matrix and performing at least a partial SVD, which scales with the total column dimension N_c. Embedding such spectral computations inside a combinatorial search over partitions is prohibitive in both time and memory.

For these reasons, our goal is not global optimality of (2.5), but rather the design of *provably safe local decisions* that guarantee feasibility of the error constraint. This perspective naturally leads to greedy clustering strategies driven by certified merge rules. The key algorithmic primitive underlying our method is a *merge certificate* that allows one to decide whether two groups of matrices can be safely merged *without explicitly computing* the singular values of the concatenated matrix.

Definition 2.10 (Compression-aware merge certificate)**.** Let $M = [M_1, M_2]$ denote the concatenation of two matrices. A compression-aware merge certificate is a computable condition $\mathcal{C}(M_1, M_2, r, \varepsilon)$ such that

$$\mathcal{C}(M_1, M_2, r, \varepsilon) \implies \mathcal{E}_r(M) = \left(\sum_{j>r} \sigma_j^2(M) \right)^{1/2} \leq \varepsilon,$$

without explicitly computing the singular values $\{\sigma_j(M)\}$.

Such certificates enable greedy clustering schemes in which clusters are merged only when the resulting approximation error is *provably admissible*. Importantly, this shifts the computational burden from repeated large-scale SVDs to inexpensive spectral surrogates and bounds. This philosophy mirrors *safe screening rules* in sparse optimization, where variables are elimi-

nated or grouped based on certificates that preserve optimality or feasibility, without solving the full problem [83, 84].

We now develop our compression-aware clustering methodology. We begin with a fast but coarse upper bound on the SVD compression error for concatenated matrices, derived from Weyl-type monotonicity, and use it to construct an efficient clustering procedure. In the following subsections, we refine this bound using residuals and incremental SVD updates.

2.2.1. Weyl-based upper bound for concatenated SVD compression. Our first result provides a simple *global* upper bound on the optimal rank-r SVD compression error of a concatenated matrix in terms of the individual blocks. It relies only on Frobenius norms and leading singular values of the blocks and is therefore inexpensive to evaluate.

Theorem 2.11 (Upper bound for SVD compression of a concatenated matrix)**.** *Let $A_j \in \mathbb{R}^{m\times n_j}$ for $j = 1, \dots, K$ and set*

$$M \;=\; \begin{bmatrix} A_1 & A_2 & \cdots & A_K \end{bmatrix} \in \mathbb{R}^{m\times(n_1+\cdots+n_K)}.$$

Denote by $\sigma_1(\cdot) \ge \sigma_2(\cdot) \ge \cdots$ the singular values of a matrix, and define the optimal rank-r approximation error in the Frobenius norm by

$$\mathcal{E}_r^2(M) \;:=\; \min_{\operatorname{rank}(X)\le r} \|M - X\|_F^2 \;=\; \sum_{i>r} \sigma_i^2(M).$$

Then, for every $r \ge 1$,

$$\mathcal{E}_r^2(M) \;\le\; \sum_{j=1}^{K} \|A_j\|_F^2 \;-\; \max_{1\le j\le K} \sum_{i\le r} \sigma_i^2(A_j). \tag{2.6}$$

In particular, if $r \ge \max_{1\le j\le K} \operatorname{rank}(A_j)$, then

$$\mathcal{E}_r^2(M) \;\le\; \sum_{j=1}^{K} \|A_j\|_F^2 \;-\; \max_{1\le j\le K} \|A_j\|_F^2.$$

Proof. Write the singular values of a matrix X in the non-increasing order as $\sigma_1(X) \geq \cdots \geq \sigma_{\min(m,n)}(X)$. By the Eckart–Young–Mirsky theorem (Theorem 1.7),

$$\mathcal{E}_r^2(M) = \sum_{i>r} \sigma_i^2(M) = \|M\|_F^2 - \sum_{i=1}^{r} \sigma_i^2(M).$$

Since $M = [A_1, \ldots, A_K]$ is a horizontal concatenation, its Frobenius norm decomposes to

$$\|M\|_F^2 = \sum_{j=1}^{K} \|A_j\|_F^2.$$

Thus

$$\mathcal{E}_r^2(M) = \sum_{j=1}^{K} \|A_j\|_F^2 - \sum_{i=1}^{r} \sigma_i^2(M). \tag{2.7}$$

We now relate the singular values of M to those of the blocks A_j. Define

$$B_j := A_j A_j^\top \succeq 0, \qquad j = 1, \ldots, K,$$

so that

$$MM^\top = \sum_{j=1}^{K} A_j A_j^\top = \sum_{j=1}^{K} B_j.$$

Fix $k \in \{1, \ldots, K\}$ and write

$$MM^\top = B_k + C_k, \qquad C_k := \sum_{\substack{j=1 \\ j \neq k}}^{K} B_j \succeq 0.$$

By the Weyl monotonicity (Theorem 1.9), if H, G are Hermitian with $G \succeq 0$ and $\lambda_1(\cdot) \geq \cdots \geq \lambda_n(\cdot)$ denote eigenvalues in nonincreasing order, then

$$\lambda_i(H + G) \;\geq\; \lambda_i(H), \qquad i = 1, \ldots, n.$$

Applying this with $H = B_k$ and $G = C_k$ yields

$$\lambda_i(MM^\top) = \lambda_i(B_k + C_k) \;\geq\; \lambda_i(B_k), \qquad i = 1, \ldots, m.$$

Using $\sigma_i^2(X) = \lambda_i(XX^\top)$, we obtain

$$\sigma_i^2(M) = \lambda_i(MM^\top) \;\geq\; \lambda_i(B_k) = \sigma_i^2(A_k), \qquad i = 1, \ldots, \min(m, n_k).$$

Summing over $i = 1, \ldots, r$ gives

$$\sum_{i=1}^{r} \sigma_i^2(M) \;\geq\; \sum_{i=1}^{r} \sigma_i^2(A_k) \qquad \text{for every } k = 1, \ldots, K.$$

Hence

$$\sum_{i=1}^{r} \sigma_i^2(M) \;\geq\; \max_{1 \leq j \leq K} \sum_{i=1}^{r} \sigma_i^2(A_j). \tag{2.8}$$

Combining (2.7) and (2.8), we obtain

$$\mathcal{E}_r^2(M) = \sum_{j=1}^{K} \|A_j\|_F^2 - \sum_{i=1}^{r} \sigma_i^2(M) \;\leq\; \sum_{j=1}^{K} \|A_j\|_F^2 - \max_{1 \leq j \leq K} \sum_{i=1}^{r} \sigma_i^2(A_j),$$

which is exactly (2.6). If $r \geq \operatorname{rank}(A_j)$ for all j, then $\sum_{i=1}^{r} \sigma_i^2(A_j) = \|A_j\|_F^2$, which yields the stated special case. □

Remark 2.12 (Single-anchor nature of the Weyl-based bound)**.** The Weyl-based upper bound in Theorem 2.11 relies on a *single anchor block* through the term

$$\max_j \sum_{i \leq r} \sigma_i^2(A_j),$$

and therefore cannot accumulate shared low-rank structure across multiple blocks. As a consequence, when compression arises from collective subspace alignment rather than dominance of a single matrix, the bound may substantially overestimate the true rank-r approximation error. This explains the conservative behavior of the max-norm clustering algorithm observed in subsection 2.2.4.

A concrete example illustrating worst-case looseness is given.

Example 2.13 (Perfect compressibility with a loose Weyl bound)**.** Consider rank-one matrices

$$A_j = u\, s_j\, v_j^\top, \qquad \|u\|_2 = 1,$$

sharing the same left singular vector u, with arbitrary right singular vectors v_j and scalars $s_j > 0$. The concatenated matrix $M = [A_1, \dots, A_K]$ has rank one, and hence

$$\mathcal{E}_1^2(M) = 0.$$

However, Theorem 2.11 yields

$$\mathcal{E}_1^2(M) \;\le\; \sum_{j=1}^{K} s_j^2 \;-\; \max_j s_j^2,$$

which grows with K unless a single block dominates. Thus, even in a maximally compressible setting, the Weyl-based bound can significantly overestimate the true reconstruction error.

Interpretation for clustering.. For any subset of indices $C \subseteq \{1, \dots, K\}$, let

$$M_C := [A_j]_{j \in C}$$

denote the concatenation of matrices in that cluster. Applying Theorem 2.11 to M_C yields

$$\mathcal{E}_r^2(M_C) \;\le\; \sum_{j \in C} \|A_j\|_F^2 \;-\; \max_{j \in C} \sum_{i=1}^{r} \sigma_i^2(A_j).$$

In the common regime where r is at least the rank of each block in the cluster (or large enough to recover each block essentially exactly when compressed alone), this simplifies to

$$\mathcal{E}_r^2(M_C) \;\le\; \sum_{j \in C} \|A_j\|_F^2 \;-\; \max_{j \in C} \|A_j\|_F^2 \;=\; \sum_{j \in C \setminus \{j^\star\}} \|A_j\|_F^2,$$

Algorithm 1 Weyl-based max-norm clustering

Require: Blocks $\{A_j\}_{j=1}^{K}$, tolerance ε, width budget r_{target}
Ensure: Clusters $\mathcal{C}$

1: Sort blocks by decreasing Frobenius norm
2: **while** blocks remain **do**
3: Choose the largest norm remaining block as anchor
4: Form a head by concatenating the largest norm blocks up to width r_{target}
5: Add smallest norm remaining blocks while relative tail energy $\leq \varepsilon$
6: Output the resulting cluster and remove its blocks
7: **end while**

where $j^\star$ is any index achieving the maximum Frobenius norm in the cluster. Thus, under this condition, the compression error of the concatenated cluster is bounded by the *sum of squared Frobenius norms of all blocks except the dominant one*. Intuitively, one large "anchor" matrix can absorb several smaller matrices at negligible additional error, as long as their total energy remains small.

This observation suggests a simple greedy strategy: for a given error tolerance $\varepsilon > 0$, we may form clusters by (i) selecting a high-energy block as an anchor, and (ii) attaching the lowest-energy remaining matrices as long as the sum of their squared norms does not exceed ε^2.

We introduce a lightweight clustering heuristic that relies only on Frobenius norms and the simplified Weyl-type bound from Theorem 2.11. The user specifies a tolerance $\varepsilon > 0$ that controls the *relative* rank-r approximation error permitted within each cluster.

Assuming the target rank satisfies $r \geq \max_{j \in C} \operatorname{rank}(A_j)$, the bound guarantees that every cluster C produced by Algorithm 1 obeys the relative error constraint

$$\frac{\mathcal{E}_r(M_C)}{\|M_C\|_F} \;\leq\; \varepsilon.$$

The algorithm runs in $O(K \log K)$ time, dominated by sorting the block norms, plus the cost of computing $\|A_j\|_F$, which is typically negligible compared to any SVD-based compression routine. In subsection 2.2.4 we show

that, despite its simplicity and coarse (norm-only) nature, this Weyl-based clustering already yields strong compression performance in practice.

2.2.2. Residual-based global bounds and clustering. The Weyl-based bound in subsection 2.2.1 depends only on individual blocks and is extremely fast to compute, but can be loose when the target rank r exceeds the rank of each block. We now derive a sharper global bound based on the singular values of *incremental residuals*. This bound captures how each block contributes new directions beyond the span of all previously seen blocks.

Theorem 2.14 (Global incremental lower bound on singular values)**.** *Let*

$$M_K = \begin{bmatrix} A_1 & A_2 & \cdots & A_K \end{bmatrix} \in \mathbb{R}^{m\times(n_1+\cdots+n_K)}$$

be constructed incrementally from block matrices $A_i \in \mathbb{R}^{m\times n_i}$. Define $M_0 := 0$ and let Q_{i-1} be any matrix with orthonormal columns spanning range(M_{i-1})*. For each block A_i, define its orthogonal residual*

$$R_i := (I - Q_{i-1}Q_{i-1}^\top)A_i.$$

Let $\widehat{R} \in \mathbb{R}^{m\times(n_1+\cdots+n_K)}$ be the block concatenation of all residuals,

$$\widehat{R} := \begin{bmatrix} R_1 & R_2 & \cdots & R_K \end{bmatrix},$$

and let $\mu_1 \geq \mu_2 \geq \cdots \geq 0$ denote the singular values of $\widehat{R}$ (in non-increasing order, extended by zeros when necessary).

Then, for every $j \geq 1$,

$$\sigma_j(M_K) \;\geq\; \mu_j.$$

In particular, as new blocks are added, the sequence of lower bounds $\{\mu_j\}_{j\geq 1}$ is monotone non-decreasing and incorporates the contributions of all residual components discovered during the incremental construction.

Proof. We prove the result by comparing the Gram matrix of M_K with the Gram matrix of the concatenated residuals $\widehat{R}$, and then invoking the eigenvalue monotonicity for Hermitian matrices.

Step 1: Gram matrix of the final concatenation.. Define the (symmetric, positive semidefinite) Gram matrix

$$X := M_K M_K^\top = \begin{bmatrix} A_1 & A_2 & \cdots & A_K \end{bmatrix} \begin{bmatrix} A_1 & A_2 & \cdots & A_K \end{bmatrix}^\top = \sum_{i=1}^{K} A_i A_i^\top,$$

where the singular values of M_K are related to the eigenvalues of X,

$$\sigma_j^2(M_K) = \lambda_j(X), \qquad j = 1, \ldots, m,$$

and the eigenvalues are ordered non-increasingly: $\lambda_1(X) \geq \lambda_2(X) \geq \cdots \geq \lambda_m(X) \geq 0$.

Step 2: Decomposition of each block into "old span + residual".. By construction, Q_{i-1} has orthonormal columns spanning range(M_{i-1}). We can therefore write each block A_i as a sum of a part in the span of Q_{i-1} and an orthogonal residual. More precisely, define

$$B_i := Q_{i-1}^\top A_i \in \mathbb{R}^{r_{i-1} \times n_i},$$

where $r_{i-1} = \text{rank}(M_{i-1}) = \text{cols}(Q_{i-1})$, and recall that

$$R_i := (I - Q_{i-1} Q_{i-1}^\top) A_i.$$

Then we have the orthogonal decomposition

$$A_i = Q_{i-1} B_i + R_i,$$

with

$$\text{range}(Q_{i-1} B_i) \subseteq \text{range}(Q_{i-1}) = \text{range}(M_{i-1}), \quad \text{range}(R_i) \subseteq \text{range}(M_{i-1})^\perp.$$

In particular, $Q_{i-1} B_i$ and R_i have orthogonal column spaces, therefore, their Gram matrices have no cross terms:

$$(Q_{i-1} B_i)(R_i)^\top = Q_{i-1} B_i R_i^\top = 0, \qquad R_i (Q_{i-1} B_i)^\top = 0.$$

Using the decomposition $A_i = Q_{i-1}B_i + R_i$, we expand

$$A_iA_i^\top = (Q_{i-1}B_i + R_i)(Q_{i-1}B_i + R_i)^\top = Q_{i-1}B_iB_i^\top Q_{i-1}^\top + R_iR_i^\top,$$

since the cross terms vanish by the orthogonality noted above. Therefore,

$$A_iA_i^\top - R_iR_i^\top = Q_{i-1}B_iB_i^\top Q_{i-1}^\top \succeq 0, \tag{2.9}$$

i.e., $A_iA_i^\top \succeq R_iR_i^\top$ in the Loewner (positive semidefinite) order.

Summing (2.9) over $i = 1, \ldots, K$ gives

$$\sum_{i=1}^{K} A_iA_i^\top \succeq \sum_{i=1}^{K} R_iR_i^\top.$$

By the definition,

$$X = M_KM_K^\top \succeq Y := \sum_{i=1}^{K} R_iR_i^\top. \tag{2.10}$$

Step 3: Orthogonality of the residual ranges.. We now show that the column spaces of the residuals R_i are mutually orthogonal. This is a consequence of the incremental construction.

Since each R_j is a linear combination of the columns of A_j, we have $\text{range}(R_j) \subseteq \text{range}(A_j)$. Because $\text{range}(M_{i-1})$ contains $A_1, \ldots, A_{i-1}$, it follows that

$$\text{range}(R_j) \subseteq \text{range}(M_{i-1}), \qquad j < i.$$

On the other hand,

$$R_i = (I - Q_{i-1}Q_{i-1}^\top)A_i$$

lies in the orthogonal complement of the $\text{range}(Q_{i-1})$, which is equal to the orthogonal complement of $\text{range}(M_{i-1})$. Thus

$$\text{range}(R_i) \subseteq \text{range}(M_{i-1})^\perp \quad \Rightarrow \quad \text{range}(R_i) \perp \text{range}(R_j) \text{ for all } j < i.$$

By symmetry of the indices, this shows that the subspaces $\text{range}(R_1), \ldots, \text{range}(R_K)$ are pairwise orthogonal.

Step 4: Gram matrix of the concatenated residuals.. Consider the concatenated residual matrix

$$\widehat{R} := \begin{bmatrix} R_1 & R_2 & \cdots & R_K \end{bmatrix} \quad \text{and} \quad \widehat{R}\,\widehat{R}^\top = \sum_{i=1}^{K} R_i R_i^\top = Y.$$

Because the column spaces of R_i are pairwise orthogonal, there exists an orthonormal basis of $\mathbb{R}^m$ in which Y becomes block diagonal with blocks $R_i R_i^\top$ and possibly an additional zero block (if the sum of their ranks is $< m$). In such a basis, the eigenvalues of Y are just the multiset union of the eigenvalues of $R_i R_i^\top$, i.e., the squared singular values of each R_i.

Equivalently, if $\mu_1 \geq \mu_2 \geq \cdots \geq 0$ are the singular values of $\widehat{R}$ (padded by zeros when necessary), then

$$\lambda_j(Y) = \mu_j^2, \qquad j = 1, \ldots, m.$$

Step 5: Eigenvalue monotonicity and conclusion.. From (2.10), $X \succeq Y$, i.e., $X - Y$ is positive semidefinite. By Weyl's monotonicity theorem (see Theorem 1.9), the eigenvalues of X and Y satisfy

$$\lambda_j(X) \;\geq\; \lambda_j(Y), \qquad j = 1, \ldots, m.$$

Combining this with the identities

$$\lambda_j(X) = \sigma_j^2(M_K), \qquad \lambda_j(Y) = \mu_j^2,$$

we obtain

$$\sigma_j^2(M_K) \;\geq\; \mu_j^2, \qquad j = 1, \ldots, m.$$

Since both sides are nonnegative, taking square roots preserves the inequality:

$$\sigma_j(M_K) \;\geq\; \mu_j, \qquad j = 1, \ldots, m.$$

This is exactly the claimed bound. □

This theorem says that the singular values of the concatenated matrix M_K are bounded from below by the singular values of the concatenated residuals $\widehat{R}$. Intuitively, every time a block contributes a component outside the span of all previously seen blocks, this "new direction" permanently lifts the spectrum of M_K.

Corollary 2.15 (Global upper bound on optimal SVD compression error)**.** *In the setting of Theorem 2.14, let M_K and μ_j be as above. For any target rank $r \geq 0$, define the optimal rank-r SVD compression error of M_K in the Frobenius norm by*

$$\mathcal{E}_r(M_K) \;:=\; \min_{\operatorname{rank}(X)\leq r} \|M_K - X\|_F \;=\; \left(\sum_{j>r} \sigma_j(M_K)^2\right)^{1/2},$$

where $\sigma_1(M_K) \geq \sigma_2(M_K) \geq \cdots \geq 0$ are the singular values of M_K.

Then, for every $r \geq 0$,

$$\mathcal{E}_r^2(M_K) \;\leq\; \sum_{i=1}^{K} \|A_i\|_F^2 \;-\; \sum_{j=1}^{r} \mu_j^2. \tag{2.11}$$

In particular, as new blocks A_i are added and $\widehat{R}$ accumulates more residual components, the quantity

$$\sum_{i=1}^{K} \|A_i\|_F^2 \;-\; \sum_{j=1}^{r} \mu_j^2$$

provides a global, monotonically decreasing *upper bound on the best achievable rank-r approximation error for M_K, computable from the per-block Frobenius norms and the singular values (or estimates) of $\widehat{R}$.*

Proof. By Theorem 2.14, $\sigma_j(M_K) \geq \mu_j$ for all $j \geq 1$, hence

$$\sum_{j=1}^{r} \sigma_j^2(M_K) \;\geq\; \sum_{j=1}^{r} \mu_j^2.$$

The Frobenius norm of M_K satisfies

$$\|M_K\|_F^2 = \sum_{j\geq 1} \sigma_j^2(M_K) = \sum_{i=1}^{K} \|A_i\|_F^2,$$

since M_K is the horizontal concatenation of the blocks A_i. By Theorem 1.7,

$$\mathcal{E}_r^2(M_K) = \sum_{j>r} \sigma_j^2(M_K) = \|M_K\|_F^2 - \sum_{j=1}^{r} \sigma_j^2(M_K).$$

Substituting $\|M_K\|_F^2 = \sum_{i=1}^{K} \|A_i\|_F^2$ and using the lower bound on the top-r energy yields

$$\mathcal{E}_r^2(M_K) = \sum_{i=1}^{K} \|A_i\|_F^2 - \sum_{j=1}^{r} \sigma_j^2(M_K) \;\leq\; \sum_{i=1}^{K} \|A_i\|_F^2 - \sum_{j=1}^{r} \mu_j^2,$$

which is exactly (2.11). □

To clarify when the residual-based bound of Corollary 2.15 is exact, informative, or potentially conservative, we characterize its behavior under different structural regimes of the concatenated blocks.

Proposition 2.16 (Exactness and degeneracy of the residual-based bound)**.** *Let*

$$M_K = \begin{bmatrix} A_1 & A_2 & \cdots & A_K \end{bmatrix} \in \mathbb{R}^{m\times(n_1+\cdots+n_K)}$$

be formed by horizontal concatenation, and let $R_1, \ldots, R_K$ be the incremental residuals defined as in Theorem 2.14. Let $\mu_1 \geq \mu_2 \geq \cdots \geq 0$ denote the singular values of the concatenated residual matrix $\widehat{R} := [R_1, \ldots, R_K]$.

Exactness under orthogonal subspace growth. *If the residual subspaces* $\mathrm{range}(R_1), \ldots, \mathrm{range}(R_K)$ *are mutually orthogonal and*

$$\mathrm{range}(M_K) = \bigoplus_{i=1}^{K} \mathrm{range}(R_i),$$

then the singular values of M_K coincide with those of $\widehat{R}$, i.e.

$$\sigma_j(M_K) = \mu_j \quad \text{for all } j,$$

and the residual-based bound in Corollary 2.15 holds with equality for every target rank r.

Degeneracy under nested column spaces. *If*

$$\operatorname{range}(A_1) \supseteq \operatorname{range}(A_2) \supseteq \cdots \supseteq \operatorname{range}(A_K),$$

then $R_i = 0$ for all $i \geq 2$, and the residual-based bound reduces to

$$\mathcal{E}_r^2(M_K) \leq \sum_{i=2}^{K} \|A_i\|_F^2,$$

which may be arbitrarily loose when all blocks lie in a common low-dimensional subspace.

Proof. **Exactness.** Under the stated assumptions, the residual subspaces $\operatorname{range}(R_1), \ldots, \operatorname{range}(R_K)$ are mutually orthogonal and together span $\operatorname{range}(M_K)$. Consequently, the Gram matrix of the concatenated residuals satisfies

$$\widehat{R}\widehat{R}^\top = \sum_{i=1}^{K} R_i R_i^\top = M_K M_K^\top.$$

Thus the eigenvalues of $M_K M_K^\top$ coincide with those of $\widehat{R}\widehat{R}^\top$, implying $\sigma_j(M_K) = \mu_j$ for all j. Substituting into the definition of the optimal rank-r approximation error yields equality in the bound of Corollary 2.15.

Degeneracy. If $\operatorname{range}(A_i) \subseteq \operatorname{range}(M_{i-1})$, then by definition of the residual

$$R_i = (I - Q_{i-1}Q_{i-1}^\top)A_i = 0.$$

Under the nesting assumption, this holds for all $i \geq 2$. The claimed bound then follows immediately from Corollary 2.15 by noting that $\mu_j = 0$ for all $j > \operatorname{rank}(A_1)$.

□

Remark 2.17 (Near-tightness under weak subspace overlap)**.** The residual-based bound of Corollary 1 becomes informative whenever each appended block contributes a non-negligible component outside the span of previously concatenated matrices. In such regimes, the leading singular values of the residual concatenation $\widehat{R}$ closely track those of the full matrix M_K, and the resulting upper bound on the truncated SVD error is empirically tight.

Conversely, when newly appended blocks lie largely within an already spanned subspace, residual energies are small and the bound may become conservative. This behavior reflects a fundamental limitation of worst-case guarantees based solely on subspace innovation and is intrinsic to concatenation-aware compression.

Interpretation for clustering.. For a cluster $C \subseteq \{1, \ldots, K\}$, let $M_C := [A_j]_{j \in C}$ denote its concatenated matrix and construct residuals R_j incrementally as the blocks in C are appended. Let $\mu_1(M_C) \geq \cdots \geq \mu_r(M_C)$ be the leading r singular values of the corresponding residual concatenation $\widehat{R}_C$. Corollary 2.15 implies

$$\mathcal{E}_r^2(M_C) \;\leq\; \sum_{j \in C} \|A_j\|_F^2 - \sum_{i=1}^{r} \mu_i^2(M_C).$$

Compared to the Weyl-based bound (subsection 2.2.1), this residual-based bound incorporates *all* new directions discovered as the cluster grows, not just the energy of the largest block. It is therefore strictly tighter whenever multiple blocks contribute nontrivial orthogonal components.

We next introduce a clustering procedure that employs the residual-based bound as its merging criterion. At each step, candidate blocks are considered in an order determined by a user-selected `sort_mode`:

- `frobenius`: candidates are ordered by increasing Frobenius norm, yielding a fast heuristic closely related to Algorithm 1;
- `residual`: candidates are ordered by increasing residual norm $\|(I - Q_C Q_C^\top) A_j\|_F$ with respect to the current cluster subspace Q_C, enabling a more geometry-aware exploration of the data at higher computational cost.

Algorithm 2 Residual-based clustering

Require: Blocks $\{A_j\}_{j=1}^{K}$, tolerance ε, target rank r
Ensure: Clusters $\mathcal{C}$

1: Sort blocks by decreasing $\|A_j\|_F$
2: **while** blocks remain **do**
3: Select the largest remaining block as anchor
4: Maintain an incremental rank-r energy estimate for the cluster
5: Add remaining blocks (by small norm or small residual) while

$$\frac{\|M_C\|_F^2 - \sum_{\ell=1}^{r} \mu_\ell^2}{\|M_C\|_F^2} \;\leq\; \varepsilon^2$$

6: Output the cluster and remove its blocks
7: **end while**

By construction and by Corollary 2.15, every cluster C produced by Algorithm 2 satisfies the relative error guarantee

$$\frac{\mathcal{E}_r(M_C)}{\|M_C\|_F} \;\leq\; \varepsilon.$$

Compared to the Weyl-based Algorithm 1, this residual-based approach incurs additional computational overhead due to incremental residual projections and singular-value updates. In return, it provides substantially tighter error control and more accurate clustering, as confirmed empirically in Section 2.2.4.

2.2.3. Approximate compression bound via incremental truncated SVD. The residual-based bound in subsection 2.2.2 yields a provable upper bound on the optimal rank-r compression error, but it only adds singular values of new directions without updating the singular values of the old ones. To obtain a more tight approximation, we maintain an orthonormal basis Q_t for the approximate dominant left singular subspace of $M_t = [A_1, \ldots, A_t]$, together with a small Gram matrix S_t whose eigenvalues track the dominant singular values. Using these approximate singular values, we define a plug-in estimator of the SVD compression error.

Corollary 2.18 (Plug-in estimator of SVD compression error from incremental singular values)**.** *Let*

$$M_K \;=\; \Big[A_1 \;\; A_2 \;\; \cdots \;\; A_K\Big] \in \mathbb{R}^{m\times(n_1+\cdots+n_K)},$$

and define the total Frobenius energy of M_K *by*

$$\|M_K\|_F^2 \;=\; \sum_{i=1}^{K} \|A_i\|_F^2.$$

Let $\widetilde{\sigma}_1(M_K) \geq \widetilde{\sigma}_2(M_K) \geq \cdots \geq 0$ *denote the approximate singular values produced by the truncated incremental scheme, i.e. the square roots of the retained eigenvalues of the Gram matrix in the compressed basis.*

For any target rank $r \geq 0$*, define the optimal rank-r SVD compression error*

$$\mathcal{E}_r(M_K) \;:=\; \min_{\operatorname{rank}(X)\leq r} \|M_K - X\|_F \;=\; \left(\sum_{j>r} \sigma_j^2(M_K)\right)^{1/2},$$

where $\sigma_1(M_K) \geq \sigma_2(M_K) \geq \cdots \geq 0$ *are the true singular values of* M_K*. Then the quantity*

$$\widetilde{\mathcal{E}}_r(M_K) \;:=\; \left(\sum_{i=1}^{K} \|A_i\|_F^2 \;-\; \sum_{j=1}^{r} \widetilde{\sigma}_j^2(M_K)\right)^{1/2}$$

is a natural plug-in estimator of $\mathcal{E}_r(M_K)$*. In particular, if the incremental scheme is executed without any truncation (so that* $\widetilde{\sigma}_j(M_K) = \sigma_j(M_K)$ *for all* j*), then*

$$\widetilde{\mathcal{E}}_r(M_K) \;=\; \mathcal{E}_r(M_K).$$

Proof. By definition of the Frobenius norm and the block structure of M_K, we have

$$\|M_K\|_F^2 = \sum_{i=1}^{K} \|A_i\|_F^2.$$

On the other hand, for the true singular values $\sigma_1(M_K) \geq \sigma_2(M_K) \geq \cdots$, the Eckart–Young–Mirsky theorem (Theorem 1.7) gives

$$\mathcal{E}_r^2(M_K) = \sum_{j>r} \sigma_j^2(M_K) = \sum_{j\geq 1} \sigma_j^2(M_K) - \sum_{j=1}^{r} \sigma_j^2(M_K) = \|M_K\|_F^2 - \sum_{j=1}^{r} \sigma_j^2(M_K).$$

The plug-in estimator $\widetilde{\mathcal{E}}_r(M_K)$ is obtained by replacing the unknown true singular values $\sigma_j(M_K)$ in this identity with their incremental approximations $\widetilde{\sigma}_j(M_K)$:

$$\widetilde{\mathcal{E}}_r^2(M_K) := \|M_K\|_F^2 - \sum_{j=1}^{r} \widetilde{\sigma}_j^2(M_K) = \sum_{i=1}^{K} \|A_i\|_F^2 - \sum_{j=1}^{r} \widetilde{\sigma}_j^2(M_K).$$

If the incremental scheme is run without truncation, then by Corollary 1.12 the approximated singular values coincide with the true ones, $\widetilde{\sigma}_j(M_K) = \sigma_j(M_K)$ for all j, and therefore

$$\widetilde{\mathcal{E}}_r^2(M_K) = \|M_K\|_F^2 - \sum_{j=1}^{r} \sigma_j^2(M_K) = \mathcal{E}_r^2(M_K),$$

which implies $\widetilde{\mathcal{E}}_r(M_K) = \mathcal{E}_r(M_K)$. In the truncated case, $\widetilde{\mathcal{E}}_r(M_K)$ is an approximation to $\mathcal{E}_r(M_K)$ obtained by this plug-in substitution. □

Interpretation and limitations.. The estimator $\widetilde{\mathcal{E}}_r(M_K)$ should be interpreted as a data-driven proxy for the true truncation error rather than a certified bound. When the incremental subspace remains well aligned with the true top-r singular subspace (e.g., under a persistent spectral gap or when newly appended blocks contribute low-energy components orthogonal to the dominant directions) the estimator is empirically tight. In contrast, when truncation discards directions that later reappear with significant energy, the estimator may temporarily underestimate the true error. This limitation is intrinsic to all truncated incremental SVD schemes and motivates the separation between provable and approximate clustering strategies.

Algorithm 3 Approximate residual-based clustering

Require: Blocks $\{A_j\}_{j=1}^{K}$, tolerance ε, target rank r

Ensure: Clusters $\mathcal{C}$

1: Sort blocks by decreasing $\|A_j\|_F$

2: **while** blocks remain **do**

3: Select the largest remaining block as anchor

4: Initialize a cluster subspace model that tracks the leading r singular values approximately

5: Add remaining blocks (by small norm or small residual) while

$$\frac{\|M_C\|_F^2 - \sum_{\ell=1}^{r} \widetilde{\sigma}_\ell^2(M_C)}{\|M_C\|_F^2} \;\le\; \varepsilon^2$$

where $\sum_{\ell=1}^{r} \widetilde{\sigma}_\ell^2(M_C)$ is an incremental approximation

6: Output the cluster and remove its blocks

7: **end while**

The Algorithm 3 replaces the exact residual-based error bound with a plug-in estimator $\widetilde{\mathcal{E}}_r(\cdot)$ obtained from an incremental truncated SVD of the concatenated cluster. Structurally, it mirrors Algorithm 2: blocks are considered in a fixed order, and new blocks are added to the current cluster until the (approximate) relative rank-r compression error exceeds the prescribed tolerance ε.

The key distinction is that, instead of explicitly tracking residual singular values, the algorithm maintains an incremental low-rank approximation of the cluster matrix and uses it directly to estimate the captured rank-r energy. This design targets the large-scale regime: each merge step operates only on small matrices of size $O(r)$, independent of the number of blocks already assigned to the cluster.

As in Algorithm 2, two ordering strategies are supported: `frobenius`, which prioritizes candidates with smaller Frobenius norm $\|A_j\|_F$, and `residual`, which orders candidates by the norm of the projected residual $(I - Q_C Q_C^\top)A_j$ with respect to the current cluster subspace Q_C.

Unlike Algorithms 1 and 2, this approximate method does not yield a

formal worst-case bound on the true compression error. Nevertheless, our experiments in subsection 2.2.4 show that the plug-in estimator $\widetilde{\mathcal{E}}_r(M_C)$ is sufficiently accurate to guide clustering decisions and achieves the most favorable trade-off between computational cost and compression quality in large-scale settings.

Dataset	# Blocks	Original Shape	Final Shape	Size (GB)
Qualcomm MIMO SCM	2468	$(2, 20, 4, 50, 32)$	$(12800, 20)$	2.35
BigEarthNet-S1	10000	$(120, 120, 2)$	$(1440, 20)$	1.07
PDEBench (Advection)	5000	$(1024, 201)$	$(3072, 67)$	3.83
SmolVLM2 256M	333760	–	$(768, 1)$	0.95

Table 2.1. Datasets used for evaluation (actual matrix shapes used in experiments).

2.2.4. Numerical experiments. We evaluate the proposed clustering–compression framework on four datasets with fundamentally different generative structure and spectral properties: (i) massive MIMO channel tensors with high-dimensional complex-valued geometry [85], (ii) Sentinel-1 SAR satellite imagery [86], (iii) PDE-generated advective flows [87], and (iv) SmolVLM2 256M model weights [88]. These domains exhibit markedly different spectrum decay rates and inter-block alignment, allowing us to assess the robustness of the proposed algorithms beyond a single application regime. All datasets are reshaped into collections of fixed-shape matrices prior to clustering. For BigEarthNet-S1 and PDEBench, we randomly sample blocks from the full datasets to obtain representative subsets of manageable size. For the Qualcomm MIMO dataset, we use a single batch of channel realizations provided by the official release. For SmolVLM2, whose weight tensors have heterogeneous shapes, we reshape each weight tensor into a fixed-length vector and treat each vector as an individual block, excluding bias parameters. The exact matrix shapes, number of blocks, and dataset statistics used in the experiments are reported in Table 2.1.

We evaluate three clustering algorithms: (i) **max-norm clustering**, derived from the Weyl-type upper bound (Theorem 2.11, Algorithm 1); (ii) **residual-based clustering**, evaluated with two sorting strategies: norm-descending and residual-ascending, based on Theorem 2.14 (Algorithm 2); (iii) **approximate incremental clustering**, also evaluated with both sorting strategies, derived from Corollary 2.18 (Algorithm 3). All algorithms depend on a target relative reconstruction error and a target approximation rank.

All experiments were run under identical hardware conditions: Intel Core i5-13600KF (20 threads) with 64 GB RAM.

Reconstruction error. Let $X \in \mathbb{R}^{M\times N}$ denote the concatenated matrix formed from all blocks assigned to a cluster, and let $\hat{X}$ be its low-rank reconstruction after clustering and decoding. We measure reconstruction error using the relative Frobenius norm error

$$\varepsilon_{\text{rel}} = \frac{\|X - \hat{X}\|_F}{\|X\|_F}. \tag{2.12}$$

For the max-norm and residual-based clustering algorithms, Theorems 2.11 and 2.14 guarantee that the relative reconstruction error does not exceed the user-specified error constraint. The approximate incremental algorithm does not provide a formal guarantee, however, no violations of the prescribed error thresholds were observed in any experiment. This empirical stability is consistent with the presence of strong spectral gaps in the tested datasets.

Compression ratio. Assume that each block has shape (m, n) and that a cluster contains K blocks approximated with target rank r. The uncompressed representation stores Kmn parameters. After compression, a shared basis requires mr parameters, while per-block coefficients require Knr parameters, resulting in a total of $r(m + Kn)$ parameters. The compression ratio is defined as the ratio between the number of parameters before and after compression.

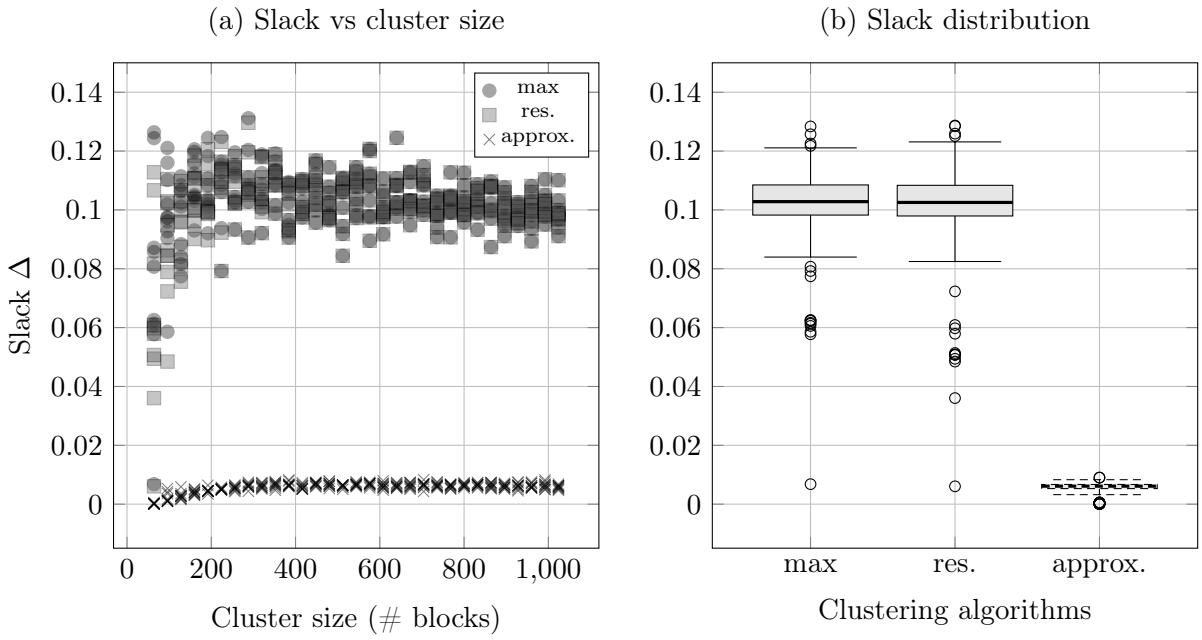


Figure 2.1. SmolVLM diagnostic of estimator conservativeness. Slack $\Delta = \widetilde{\mathcal{E}}_r - \mathcal{E}_r$ between predicted and true rank-r SVD reconstruction error. For each cluster size and estimator, all 10 independent trials (uniform block samples) are shown. (a) slack versus cluster size; (b) empirical slack distribution across estimators.

Estimator conservativeness diagnostic. In addition to end-to-end compression performance, we explicitly evaluate how conservative the proposed error estimators are relative to the true truncated SVD error. For a fixed target rank r, we uniformly sample subsets of blocks of increasing cluster size, form their concatenation, and compute the true optimal rank-r reconstruction error $\mathcal{E}_r$ via an explicit SVD. For each sampled subset, we also compute the corresponding predicted error $\tilde{\mathcal{E}}_r$ produced by the max-norm, residual-based, and approximate incremental estimators. For each cluster size and estimator, this procedure is repeated over 10 independent random trials with different block samples; *all individual trial outcomes are shown* in Figure 2.1. We report the resulting *slack* $\Delta = \tilde{\mathcal{E}}_r - \mathcal{E}_r$, which directly measures estimator conservativeness. As predicted by Theorems 2.11-2.14, the max-norm and residual-based estimators remain strictly conservative across all trials, exhibiting a consistently positive slack that is largely insensitive to cluster size. In contrast, the approximate incremental estimator produces

Method	Qualcomm	BigEarthNet	PDEBench	SmolVLM2	Wall time†
max norm	2.138	1.000*	1.004*	1.581	1×
res. (norm sorting)	2.276	1.000*	1.005*	1.588	2168×
res. (res. sorting)	2.243	1.000*	1.002*	1.547	4620×
approx. (norm sorting)	2.301	1.789	**45.434**	1.642	**100**×
approx. (res. sorting)	**2.334**	**1.870**	**45.434**	**1.807**	1792×

* Clustering failed; no compression achieved (compression ratio ≈ 1.0).

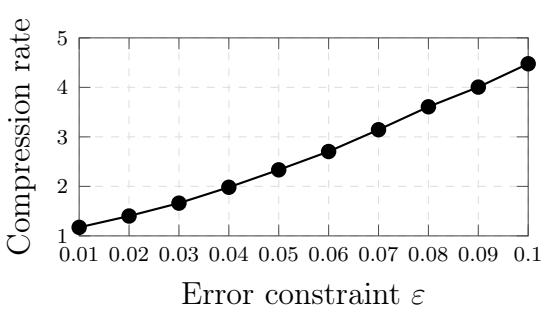
† Wall time is reported relative to a baseline of 130 ms.

Table 2.2. Compression ratios across datasets and worst-case clustering wall time. Wall time is reported relative to max-norm (higher is slower). Clustering were run with 5% relative reconstruction error constraint for Qualcomm, BigEarthNet and PDEBench and with 20% for SmolVLM. And target rank is fixed to 20 for Qualcomm and BigEarthNet, to 67 for PDEBench and to 32 for SmolVLM.

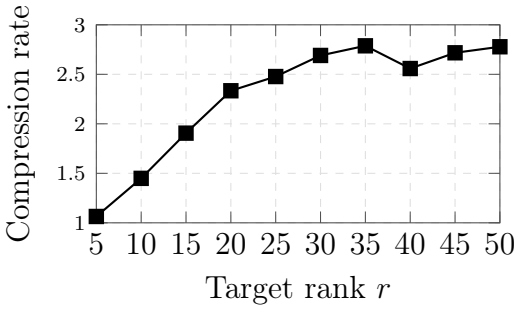
substantially smaller slack (often close to zero) indicating a much tighter but non-guaranteed estimate of the true error. The observed dispersion across trials reflects variability induced by block selection while preserving a clear qualitative separation between estimators.

Clustering results. Table 2.2 reports compression ratios and worst-case clustering wall times across all datasets. Entries marked with $*$ indicate failure to achieve compression, defined as the inability to form clusters larger than individual blocks, resulting in compression ratios close to one. This behavior is not caused by numerical instability, but arises when conservative error bounds prevent aggregation of blocks into shared low-rank subspaces.

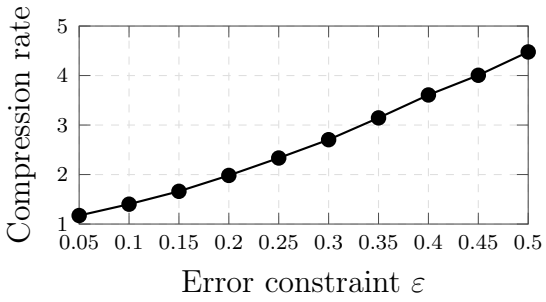
The results reveal a clear trade-off between computational efficiency, compression performance, and theoretical guarantees. Max-norm clustering is consistently the fastest method, but yields conservative clustering due to loose worst-case bounds, limiting achievable compression. Residual-based clustering enforces strict reconstruction guarantees and produces stable results when feasible, but incurs several orders of magnitude higher runtime. Approximate incremental clustering achieves the highest compression, while remaining substantially faster than exact residual-based methods.



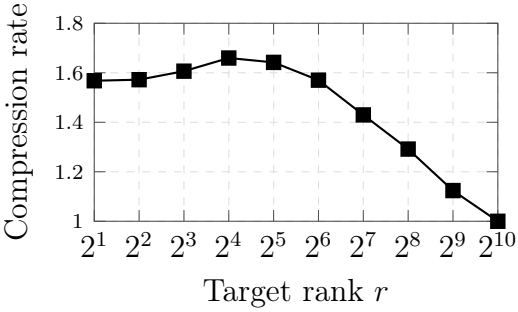
(a) Qualcomm: sweep ε, fixed $r = 20$.



(b) Qualcomm: rank sweep, fixed error 0.05.



(c) SmolVLM2: sweep ε, fixed $r = 32$.



(d) SmolVLM2: rank sweep, fixed error 0.2.

Figure 2.2. Compression rates under feasibility constraints. (a) Qualcomm dataset: fixed target rank and varying error constraint. (b) Qualcomm dataset: fixed error constraint and varying target rank. (c) SmolVLM2 model weights: fixed target rank and varying error constraint. (d) SmolVLM2 model weights: fixed error constraint and varying target rank.

Hyperparameter sensitivity. We analyze the dependence of compression performance on the target reconstruction error and target rank. Figure 2.2 shows results for the Qualcomm MIMO dataset using approximate clustering with residual-based sorting, and reports results for SmolVLM2 weights using approximate clustering with norm-based sorting. Across both datasets, compression increases approximately linearly with the allowed reconstruction error, consistent with low-rank approximation theory. In contrast, the dependence on target rank is non-monotonic: increasing rank does not necessarily improve compression. This behavior reflects the interaction between rank selection and inter-block subspace alignment, highlighting that rank is not merely a capacity parameter but must be chosen carefully to balance expressivity and shared structure.

Method	Compression	Rel. error
Random clustering (K=10)	178.1	0.704 ± 0.034
Random clustering (K=100)	23.8	0.363 ± 0.110
Random clustering (K=1000)	2.46	0.199 ± 0.134
k-means (K=10)	178.3	0.886
k-means (K=100)	23.8	0.322 ± 0.343
k-means (K=1000)	2.46	0.881 ± 0.098
HDBSCAN (min cluster size = 2)	–	–
Ours (approx. residual)	2.334	**0.044 ± 0.006**

Table 2.3. Classical clustering baselines and proposed approximate method on the Qualcomm dataset.

Classical clustering baselines. We additionally compare against baseline clustering methods on the Qualcomm dataset. Table **??** reports results for random clustering and k-means with varying numbers of clusters and for HDBSCAN. These methods are fundamentally misaligned with the concatenated SVD compression objective. Classical clustering algorithms optimize geometric distortion in the ambient space and provide no mechanism to control spectral reconstruction error after low-rank decoding. As a result, high compression is achieved only at the cost of unacceptable and highly unstable reconstruction error, while density-based clustering fails entirely by labeling all points as outliers. This confirms that standard clustering techniques are unsuitable for controlled low-rank compression of concatenated matrices.

2.3. Conclusion

This chapter established a spectral and algorithmic framework for deciding when matrices should be compressed jointly via concatenated SVD and when they should be kept separate. The central contribution is to replace heuristic grouping with a 'compression-aware' criterion: candidate groups are accepted only when their predicted low-rank reconstruction error satisfies an explicit feasibility constraint. By linking singular-value perturbation

analysis to truncation error control, we obtained practical decision rules for incremental matrix grouping with quantitative guarantees.

On the theoretical side, we analyzed how singular values change under blockwise concatenation and perturbations, and clarified the role of Gram-matrix viewpoints ($M^\top M$ versus $MM^\top$) for deriving usable bounds. These results explain why naive geometric similarity is insufficient for grouping: even small inter-block misalignment can significantly alter the effective spectrum of a concatenation and therefore its low-rank compressibility. The perturbation bounds developed in this chapter provide a principled way to track this effect as new blocks are appended.

On the algorithmic side, we developed and compared max-norm, residual-based, and approximate incremental clustering strategies. The experiments on Qualcomm MIMO and SmolVLM2 weights reveal a consistent trade-off: conservative bounds give speed but lower compression, exact residual control gives strongest reliability but high runtime, and approximate incremental methods offer the best balance in large-scale settings. We also showed that compression grows predictably with relaxed error tolerance, while dependence on target rank is non-monotonic due to subspace alignment effects.

Finally, comparisons with random clustering, k-means, and HDBSCAN demonstrate that classical clustering objectives are poorly matched to controlled low-rank decoding. In contrast, the proposed compression-aware formulation directly optimizes what matters for deployment: feasible reconstruction error under maximal compression. These results provide both theoretical justification and practical guidance for structured low-rank compression of collections of matrices, and they motivate the next chapter's extensions to broader model classes and adaptive grouping policies.

Discussion and future work.. Throughout this work, the target rank is treated as a fixed hyperparameter. In practice, the optimal choice of r depends on the interaction between truncation error and cluster formation,

and may vary across clusters. Importantly, the overall compression is generally non-monotonic in r, since changing r alters not only the approximation within each cluster but also which merges satisfy the error constraint. A principled approach to selecting r without solving the full clustering problem remains an open question. In particular, joint optimization of rank selection and clustering, with r adapting during merging, is a promising direction for future work.

The present framework operates in an offline setting where all matrices are available in advance. In many applications, however, new data arrive sequentially. Extending compression-aware clustering to an incremental setting, where each new matrix must be assigned to an existing cluster or used to initialize a new cluster based on predicted SVD compression error, is a natural and practically important direction for future work. Such an extension would require efficient per-cluster error estimation and dynamic cluster management under streaming updates.

The current framework measures reconstruction quality using the Frobenius norm. This choice is motivated by the compression objective: for truncated SVD, the Frobenius error corresponds to the total discarded spectral energy, which admits an additive interpretation across singular directions and aligns naturally with memory–distortion trade-offs. In contrast, spectral norm controls only the largest residual singular value and therefore captures worst-case directional error rather than aggregate reconstruction quality. As a result, two clusterings may exhibit similar spectral error while differing substantially in total reconstruction fidelity. Moreover, the proposed clustering criteria rely on energy-based certificates that accumulate contributions across multiple directions, which are inherently tied to Frobenius-type quantities. Extending the framework to alternative norms, such as spectral norm, would require different merge certificates that directly control the $(r+1)$-th singular value of concatenated matrices, and is left for future work.

Beyond pure reconstruction, low-rank representations are often used as intermediate features for downstream tasks. In such settings, reconstruction error may not fully capture task-relevant information. Incorporating task-aware objectives into compression-driven clustering while retaining spectral guarantees remains an open and challenging problem.

Finally, we note that the approximate incremental estimator is closely related to randomized sketching and streaming low-rank approximation methods. In principle, it could be replaced by a randomized SVD or range-finder approach, yielding $(1 + \varepsilon)$-approximate rank-r reconstructions in Frobenius norm with high probability and near-linear computational cost. In the clustering setting, this would lead to *probabilistic* merge criteria and end-to-end guarantees that hold with high probability, in contrast to the deterministic worst-case guarantees provided by the residual-based method. However, randomized approximations do not naturally yield conservative upper bounds on the residual energy, which are required to ensure safe merges under a prescribed error tolerance. Designing sketch-based clustering procedures that preserve certificate-based guarantees while improving computational efficiency is an interesting direction for future work.

Chapter 3

Analysis of model perturbations induced by pruning

This chapter develops a unified view of pruning as a structured perturbation of neural-model parameters and studies how this perturbation propagates to task-level behavior. Rather than treating pruning only as an engineering compression tool, we analyze interpretable quantities: output deviations, geometry of language representations, return degradation in reinforcement learning, and robustness margins in neural control.

The central abstraction is shared across all settings. Let a model with parameters Θ be pruned to $\widehat{\Theta} = \Theta + \delta\Theta$, where many coordinates of $\widehat{\Theta}$ are zero. Pruning quality is then analyzed through the perturbation map

$$\mathcal{E}\colon (x, \Theta, \delta\Theta) \mapsto \Delta(x) = \|F(x;\Theta) - F(x;\widehat{\Theta})\|, \tag{3.1}$$

and then lifted from pointwise errors to problem-level criteria (perplexity, clustering consistency, return, tracking quality).

Chapter roadmap.. The chapter follows a strict empirical-to-theoretical progression. Section 3.1 studies language-specific pruning for LLM compression and quantifies the role of calibration-language mismatch. Section 3.2 uses pruning-induced weight-importance patterns to construct a metric space of 106 languages. Section 3.3 derives closed-form robustness guarantees for pruned neural controllers in deterministic nonlinear dynamics. Section 3.4 extends these guarantees to nonasymptotic bounds on return degradation in reinforcement learning. Section 3.5 synthesizes contributions, limitations, and forward directions.

3.1. Language-specific pruning for efficient LLM reduction

Large Language Models (LLMs) deliver strong performance across NLP tasks, but their scale still limits practical deployment. Alongside quantization [42, 45, 46], pruning remains a critical compression mechanism because it can reduce compute and memory while preserving model quality.

Although modern pruning methods perform well in general settings [73–75], language-conditioned effects are insufficiently studied. This section focuses on Ukrainian as a representative low-resource scenario and tests whether pruning efficacy depends on the calibration language. We compare two training-free methods — SparseGPT [7] and Wanda [47] — on LLaMA [36], LLaMA 2 [37], and Mistral [38] without retraining.

Because Transformer pruning primarily targets linear layers, the methodology is architecture-agnostic and transferable to other decoder-style models. Calibration is performed on UberText 2.0 [89]; language-mismatch effects are evaluated by pruning on English c4 [90] and testing on Ukrainian. We report dense baselines and sparse variants under unstructured and NVIDIA-friendly 2:4 semi-structured sparsity at 50% pruning [76].

The section addresses three questions: (i) sensitivity to calibration-set size, (ii) comparative efficacy of SparseGPT vs. Wanda, and (iii) dependence on calibration-language identity.

3.1.1. Related work. While our work primarily focuses on training-free approaches to language model pruning, it is essential to acknowledge the existence of methods that require post-pruning retraining [75, 91]. The effectiveness of such methods is contingent on the availability and quality of training data, making it less practical for scenarios where acquiring sufficient annotated data is a formidable task.

In the context of low-resource languages such as Ukrainian, where limited annotated data poses a significant obstacle, this limitation underscores the importance of investigating training-free approaches, which mitigate the need for additional labeled data. Therefore, we focus on methods that require only a relatively small calibration dataset for efficient model pruning.

These approaches share a similar concept: assessing weight importance based on a specific metric and input calibration data, where a larger value of the importance metric indicates that the weight should be retained. The pruning process is conducted in a layer-wise manner, involving the calculation of weight importance for each layer. Subsequently, the weights are sorted, and depending on the desired sparsity level, weights with lower importance are replaced with zeros. This streamlined approach facilitates efficient pruning, even for large-scale models.

The subsequent subsections detail these methods and highlight their practicality in low-resource language settings.

SparseGPT. In recent strides towards optimizing the efficiency of Large Language Models (LLMs), SparseGPT emerges as a pioneering one-shot pruning method [7].

The foundation of SparseGPT's pruning methodology lies in the formalization of the problem through a local layer-wise reconstruction approach. It employs a pruning metric that considers the layer-wise reconstruction problem.

$$S_{ij} = \left[\frac{|W|^2}{\text{diag}\big((X^\top X + \lambda I)^{-1}\big)} \right]_{ij} \tag{3.2}$$

The weight importance metric utilized in SparseGPT, represented by equation 3.2, incorporates the Hessian matrix in the denominator, where W denotes the weights, X represents the inputs, and λ stands for the Hessian dampening factor, employed to prevent the collapse of inverse computation. This metric underscores the importance of local layer-wise information during

the pruning process. By prioritizing such information, SparseGPT ensures the preservation of accuracy levels crucial for the optimal performance of large language models.

Wanda. The approach, termed "Pruning by Weights and Activations" [47] presents an effective solution to the pruning challenge. Wanda augments the standard weight magnitude pruning metric with input activations, effectively evaluating weight importance.

$$S_{ij} = |W_{ij}| \, \|X_j\|_2. \tag{3.3}$$

The computation of weight importance in Wanda is defined by equation 3.3, where the score for each weight W_{ij} is computed as the product of its magnitude and the norm of the corresponding input feature vector X_j. Therefore, the score encapsulates the weight's importance within the context of its associated input activations.

One of the key strengths of Wanda lies in its computational efficiency and minimal memory overhead. The method can be executed in a single forward pass, making it suitable for practical implementation in large-scale language models.

SparseGPT and Wanda therefore instantiate a clear accuracy-efficiency trade-off: SparseGPT uses a richer second-order proxy, while Wanda favors a lightweight first-order surrogate. The empirical comparison below quantifies this trade-off in a language-specific setting.

3.1.2. Experimental methodology and setup. We evaluate LLaMA 7B, LLaMA 2 7B, and Mistral v0.1 7B in FP16. The primary metric is perplexity (PPL), computed as the exponentiated average negative log-likelihood over a token sequence $X = (x_0, x_1, \ldots, x_t)$:

$$\mathrm{PPL}(X) = \exp\left\{ -\frac{1}{t} \sum_{i=0}^{t} \log p_\theta(x_i \mid x_{<i}) \right\},$$

Lower PPL indicates better predictive performance.

The target language is Ukrainian. We use UberText 2.0 [89] and exclude the social subcorpus due to short-text bias. For each of the four retained subcorpora (court, fiction, news, Wikipedia), we sample 1000 texts for calibration and 50 texts for evaluation, yielding 4000 calibration and 200 evaluation samples; each sample exceeds 8192 characters.

Calibration examples are sampled with controlled random seeds. Each configuration is run with three seeds, and we report mean and standard deviation. To test language mismatch, we additionally calibrate pruning on English c4 [90] and evaluate on the same Ukrainian test set. Dense (unpruned) models are included as baselines.

All sparse models target 50% sparsity in linear layers and compare both unstructured and 2:4 semi-structured patterns.

The experimental design addresses three questions:

1. How strongly does performance depend on calibration-set size?
2. Which training-free method performs better under matched constraints?
3. How sensitive is pruning quality to calibration-language choice?

On hardware, both methods can prune 7B models on a single NVIDIA RTX 3090 within approximately one hour; required resources scale primarily with model size and context length, and are broadly comparable to inference-time requirements [7].

3.1.3. Numerical experiments. In this section, we present and discuss the outcomes of our experiments, focusing on the perplexity metric evaluated on the Ukrainian evaluation dataset with various setups for different models.

Table 3.1 illustrates perplexity values for models pruned on UberText 2.0 dataset, employing both unstructured and 2:4 semi-structured pruning configurations with 50% sparsity. Additionally, the models underwent prun-

Method	Calibration Samples	LLaMA 7B	LLaMA 2 7B	Mistral v0.1 7B
Unstructured Wanda	64	12.162 ± 0.025	11.283 ± 0.007	**9.314 ± 0.098**
	128	12.161 ± 0.012	11.278 ± 0.007	9.726 ± 0.125
	256	**12.148 ± 0.008**	11.275 ± 0.009	10.385 ± 0.038
	512	12.152 ± 0.007	**11.254 ± 0.012**	12.262 ± 0.424
2:4 Wanda	64	31.533 ± 0.169	**30.101 ± 0.406**	**29.822 ± 0.381**
	128	31.438 ± 0.348	30.177 ± 0.361	30.741 ± 0.231
	256	31.496 ± 0.327	30.651 ± 0.353	32.709 ± 0.328
	512	**31.198 ± 0.446**	30.883 ± 0.271	34.471 ± 0.704
Unstructured SparseGPT	64	10.632 ± 0.027	9.703 ± 0.013	7.109 ± 0.003
	128	10.559 ± 0.011	9.683 ± 0.028	7.095 ± 0.011
	256	10.531 ± 0.006	9.671 ± 0.015	7.085 ± 0.003
	512	**10.529 ± 0.020**	**9.652 ± 0.012**	**7.074 ± 0.004**
2:4 SparseGPT	64	13.319 ± 0.092	11.559 ± 0.082	8.582 ± 0.036
	128	13.148 ± 0.192	11.515 ± 0.072	8.551 ± 0.041
	256	13.093 ± 0.054	11.457 ± 0.035	8.497 ± 0.006
	512	**12.994 ± 0.047**	**11.379 ± 0.008**	**8.476 ± 0.031**

Table 3.1. Perplexity values of different models and different pruning configuration.

ing using diverse calibration sample sizes (64, 128, 256, 512) to examine the relationship between sample size and performance.

Analyzing the table, it could be observed that Wanda's performance appears independent of calibration set size or, perhaps, this correlation does not consistently hold across all models. This is particularly evident in the perplexity values of unstructured models, such as Mistral v0.1 7B, where the Pearson correlation between calibration set size and perplexity mean values is 0.99, and LLaMA 2 7B, where the correlation is -0.98. Conversely, all models pruned by SparseGPT exhibit a notably high negative correlation, such as for 2:4 LLaMA 7B, where the correlation is -0.9. Hence, we can assert that Wanda's performance is not necessarily dependent on the calibration data size, while SparseGPT's performance does show such dependency. This difference could be attributed to the inherent dissimilarity in the precision of importance metrics employed by each method, where Wanda utilizes a faster but less accurate metric, and SparseGPT employs a more precise but time-intensive alternative.

The Table 3.2 presents the optimal perplexity values achieved by models pruned using both unstructured and 2:4 semi-structured configurations, each with 50% sparsity, on calibration data from UberText 2.0 or c4 datasets. Additionally, the perplexity values for the dense models are included.

The analysis of the table leads to the conclusion that, among both unstructured and 2:4 semi-structured configurations, the most effective pruning method is SparseGPT when applied to the UberText 2.0 dataset, which consists of Ukrainian texts. It is also noteworthy that the superiority of the SparseGPT pruning technique becomes evident, particularly when the pruning pattern is 2:4 semi-structured.

Furthermore, the extreme variances observed in models pruned with c4 data indicate a significant dependency on randomness in the pruning process, suggesting that the outcome is less influenced by the dataset itself.

Model	LLaMA 7B	LLaMA 2 7B	Mistral v0.1 7B
Dense	8.950	8.269	6.460
Unstructured Wanda on c4	13.953 ± 0.060	13.829 ± 0.087	41.466 ± 6.314
Unstructured SparseGPT on c4	15.797 ± 0.761	15.011 ± 0.283	9.208 ± 0.086
Unstructured Wanda on UberText 2.0	12.148 ± 0.008	11.254 ± 0.012	9.314 ± 0.098
Unstructured SparseGPT on UberText 2.0	**10.529 ± 0.020**	**9.652 ± 0.012**	**7.074 ± 0.004**
2:4 Wanda on c4	52.346 ± 1.628	79.801 ± 7.338	433.940 ± 282.154
2:4 SparseGPT on c4	89.772 ± 28.306	57.460 ± 5.379	165.516 ± 90.769
2:4 Wanda on UberText 2.0	31.198 ± 0.446	30.101 ± 0.406	29.822 ± 0.381
2:4 SparseGPT on UberText 2.0	**12.994 ± 0.047**	**11.379 ± 0.008**	**8.476 ± 0.031**

Table 3.2. Perplexity values of different models and different pruning configuration.

Moreover, we analyze the memory footprint of the models before and after pruning. As shown in Table 3.3, pruning with a 50% sparsity level reduces the memory size of the models by approximately 41%. Therefore, pruning enables a significant decrease in the memory consumption of the model's parameters while preserving parameters in 16-bit floating-point format. However, achieving such a reduction in memory usage is not feasible with unstructured sparsity. To attain this reduction, we should utilize a 2:4 semi-structured sparsity pattern, which employs an efficient sparse semi-structured tensor representation.

Model	Dense	Sparse
LLaMA 7B	12.58 Gbs	7.31 Gbs
LLaMA 2 7B	12.68 Gbs	7.40 Gbs
Mistral v0.1 7B	13.99 Gbs	8.30 Gbs

Table 3.3. Memory footprint before (dense) and after (sparse) pruning with 50% sparsity level and 2:4 semi-structured sparsity configuration of different models.

Additionally, among these three models, Mistral v0.1 7B demonstrates the best pruning performance, as indicated by the lowest residual between dense and pruned perplexity values.

Therefore, SparseGPT emerges as the preferred pruning method for language-specific applications, with its performance significantly influenced by the language of the calibration dataset.

3.2. Deep language geometry from LLM weights

This second empirical study shares the same object of study as Section 3.1: multilingual decoder-style LLMs and pruning-derived internal signals. To avoid repetition, we do not restate general LLM compression background here.

Instead, we ask a complementary question: can language-conditioned pruning signals be transformed into a quantitative geometry of languages?

Building on prior evidence that internal weight activations vary with input language [92], we represent each language by a high-dimensional binary vector derived from pruning saliency. Distances between these vectors induce a Hamming metric space (X, d_h), which is then embedded via a distance-preserving map into a low-dimensional Euclidean space (Y, d_e) for analysis and visualization.

We compute these representations for 106 languages and release the code, vectors, and analysis utilities to support reproducible downstream linguistic studies.

The contributions of this section are as follows:

- We introduce a novel approach that constructs a metric space of languages using LLM weights and apply it to 106 languages, enabling automatic and data-driven measurement of linguistic distances.
- We demonstrate that the derived metric space supports meaningful clustering of languages, reflecting both historical relationships and modern linguistic features.
- We fully open-source our work along with a tool for preliminary analysis.

The goal is not to replace linguistic theory, but to provide a scalable quantitative instrument that complements expert analysis.

3.2.1. Related work. General LLM- and pruning-oriented literature is reviewed in Section 3.1. Here we summarize only the additional prior work specific to language-distance modeling. The quantification of language similarity has a rich history, beginning with early lexical approaches. Pioneering work [93] established methods for comparing languages using shared cognates, a practice later refined by [94], which employs normalized Levenshtein distances over fixed word lists. Although these lexical methods have been successfully used to construct language family trees, they are handcrafted and require manual effort to select and curate appropriate word lists and features.

Also, resources such as the World Atlas of Language Structures [95] offer comprehensive typological data that allow languages to be represented as feature vectors. Distance measures computed over these vectors have been shown to reveal groupings consistent with established genetic relationships [96, 97]. However, these methods are limited by the quality and coverage of available databases, their reliance on expert-curated features, and their inability to fully capture language-specific variations or recent evolutionary trends.

Phonological properties offer another valuable dimension for language comparison. Studies utilizing phoneme inventory data from resources like PHOIBLE [98] demonstrate that phonological distances – often measured by overlap indices such as the Jaccard similarity – can capture both genetic relationships and areal phenomena. But phonological methods need reliable phoneme lists, are affected by how sounds are written, and often miss language structure beyond sounds.

Recent deep-learning work has popularised embedding-based measures of language distance. Multilingual encoders such as mBERT [99], XLM-R [100] and LASER [101, 102] produce contextual token embeddings that implicitly encode lexical, syntactic and semantic features. LASER is trained to output a single sentence vector directly, whereas mBERT and XLM-R require a pooling step (e.g., mean pooling or the `[CLS]` token) to obtain a sentence-level embedding. When sentence embeddings are averaged over large, balanced corpora, the resulting language-level representations have proved useful for quantifying cross-lingual similarity [103]. However, because the underlying encoders operate at the token – and therefore sentence – level, their effectiveness still depends on corpus size and domain balance.

Overall, the literature on language distance metrics has evolved from classical lexicostatistical methods and handcrafted feature extraction to sophisticated neural representations. Each approach offers valuable insights into the relationships between languages, but they often suffer from labor-intensive

preprocessing, limited database coverage, or sensitivity to input variations. This motivates our approach: rather than relying on manually curated features or sentence-based embeddings, we propose an automatic, data-driven method that leverages the internal weights of modern LLMs to construct a metric space of languages.

Moreover, to best of our knowledge, no study has attempted to derive a language metric space from decoder-only LLMs. The method introduced here is therefore the first to use weight-level signals in causal transformers for measuring cross-language similarity.

3.2.2. Methodology. The main hypothesis in this work is that Large Language Models are a good choice to measure internal language structure since they are trained to model languages. Formally, this is typically framed as maximizing the log-likelihood of the observed sequence of tokens. Let $x_1, x_2, \ldots, x_T$ represent a sequence of tokens, where $x_t \in \mathcal{V}$ and $\mathcal{V}$ is the vocabulary. The objective is to maximize the likelihood of the sequence under the model's parameters θ:

$$\mathcal{L}(\theta) = \sum_{t=1}^{\top} \log p(x_t | x_1, x_2, \ldots, x_{t-1}; \theta),$$

where $p(x_t | x_1, x_2, \ldots, x_{t-1}; \theta)$ is the conditional probability of the token x_t given the previous tokens, modeled by a neural network or another probabilistic model.

Weight importance metric. SparseGPT [7] adopts this idea within an LLM pruning algorithm. They compute the importance metric S_{ij} for a layer as follows [47]:

$$S_{ij} = \left[\frac{|W|^2}{\operatorname{diag}\Big((X^\top X + \lambda I)^{-1} \Big)} \right]_{ij}. \tag{3.4}$$

As in SparseGPT, we build X *per linear sub-layer* by stacking the pre-activation hidden states of a small calibration set into an $N \times d_{\text{in}}$ matrix. For a weight matrix W the local Hessian is $H = X^\top X$, and we invert $\left(X^\top X + \lambda I\right)$ *once per layer*. Thus, equation (3.4) is simply a matrix-valued, regularised version of the scalar error-increase criterion in equation (1.3).

In our previous work, we showed that the SparseGPT algorithm provides statistically stable results across different LLMs and data subsets in a language-specific setting. Therefore, in this work, we adopt this algorithm to compute weight-importance vectors.

Rationale behind the approach. By definition, S_{ij} quantifies the importance of weight W_{ij} for a given input. In our approach, we estimate the importance of the weights for a specific language by using datasets in that language. Consequently, S_{ij} reflects the contribution of each weight to language modeling.

Assuming that the network is well-trained on language modeling, higher S scores indicate greater contribution. If two languages yield similar patterns of important weights, it suggests that they are similar in terms of language modeling characteristics.

Constructing a metric space. To derive a vector representation from the importance metric, we treat the importance scores as coordinates in a high-dimensional space. Specifically, we define the vector

$$v = \left(S^0_{00}, S^0_{01}, \ldots, S^k_{ij}, \ldots, S^l_{nm}\right) \in \mathbb{R}^N,$$

where the set $\{W^k\}_{k=0}^l$ consists of weight matrices $W^k \in \mathbb{R}^{n_k \times m_k}$ for each layer k, and N is the total number of parameters in the chosen LLM. In other words, the vector v is obtained by flattening and concatenating all the importance matrices S^k corresponding to each layer.

There are two challenges with using the raw importance matrix S to form this vector representation:

1. The importance scores are not normalized across layers, meaning that they are only meaningful within the context of a single layer.

2. The resulting vector is high-dimensional, with each dimension represented by a floating-point number (typically 16 bits), leading to large memory requirements.

To mitigate this, we propose a thresholding approach analogous to binary quantization. Specifically, we assign a value of 1 only to the most important weights by thresholding S_{ij} at its median:

$$\widehat{S}_{ij} = \mathbf{1}\big(S_{ij} > \text{median}(S)\big).$$

This binary representation requires only 1 bit per value, reducing the storage requirement substantially compared to 16-bit floating-point representations.

Let X denote the set of language vectors (one per language) of length N. We then define a metric space on X using the Hamming distance (i.e., the XOR operation) as the metric.

For $x, y \in X$ the Hamming distance is

$$d_h(x, y) \;=\; \sum_{i=1}^{N} \mathbf{1}\big[x_i \neq y_i\big],$$

where $\mathbf{1}[\cdot]$ is the indicator function.

The function d_h is non-negative, symmetric, equals 0 iff $x = y$, and satisfies the triangle inequality, therefore, (X, d_h) is a metric space.

Isometry via dimensionality reduction. Even after quantization, the binary vectors remain high-dimensional due to the large number of model parameters, making distance computations and other latent space applications computationally expensive. To address this, we construct an isometry – a transformation that preserves distances between points when mapping from one metric space to another.

Algorithm 4 Torgerson Scaling (Classical MDS)

Require: Distance matrix $D \in \mathbb{R}^{n \times n}$, $n = |X|$

Ensure: Coordinates $Y \in \mathbb{R}^{n \times d}$ representing points in d dimensions

1: $J \leftarrow I_n - \frac{1}{n}\mathbf{1}_n$ {Compute centering matrix}
2: $D^2 \leftarrow D \odot D$ {Element-wise square of D}
3: $B \leftarrow -\frac{1}{2}\, J\, D^2\, J$ {Compute Gram matrix}
4: $(\lambda, V) \leftarrow \texttt{eigh}(B)$ {Compute the eigen-decomposition of B}
5: $(\lambda, V) \leftarrow \texttt{sort}((\lambda, V))$ {Sort eigenvalues in descending order and reorder eigenvectors accordingly}
6: $d \leftarrow \#\{\lambda_i \mid \lambda_i > \epsilon\}$ {Select dimensions with significant eigenvalues ($\epsilon \approx 10^{-10}$)}
7: $L \leftarrow \mathrm{diag}(\sqrt{\lambda_1}, \sqrt{\lambda_2}, \ldots, \sqrt{\lambda_d})$
8: $V_d \leftarrow [v_1, v_2, \ldots, v_d]$
9: **return** $Y \leftarrow V_d\, L$

In our experiments, we employ different LLMs and multiple datasets. We compute the language-by-language distance matrix for each model and dataset, and then average them to obtain a robust distance measure:

$$D_{lk} \in \mathbb{R}^{|X| \times |X|}, D_{lk} = \{d_h(v_i, v_j) : v_i, v_j \in X\},$$
$$\widehat{D} = \mathbb{E}_{l \sim p_{\mathrm{LLM}}} \mathbb{E}_{k \sim p_{\mathrm{data}}}[D_{lk}] \approx \frac{1}{nm} \sum_{l=0}^{n} \sum_{k=0}^{m} D_{lk},$$

where D_{lk} is the distance matrix computed for the lth LLM and the kth dataset, n is the number of LLMs, m is the number of datasets, and $|X|$ is the number of languages.

This averaging process reduces noise and ensures that the final distances are not overly dependent on any particular dataset or model.

We then construct an isometry

$$f \colon X \to Y,$$

where Y is a metric space endowed with the Euclidean metric $d_e(x, y) = \|x - y\|_2$.

To build f, we apply Torgerson scaling (classical multidimensional scaling) [104]. The result is a set of points $Y \in \mathbb{R}^{|X| \times d}$, where d is the minimum

number of dimensions required to preserve the distances in $\widehat{D}$ (see Algorithm 4). Notably, d is much smaller than the original dimensionality N of the language vectors and satisfies $d \leq |X|$.

Thus, the pipeline converts LLM weight-importance signals into a compact metric representation suitable for quantitative cross-language analysis.

3.2.3. Results. We evaluate the resulting metric space using clustering, visualization, and comparison against established linguistic taxonomies [105]. For clustering, we use HDBSCAN [106] and k-means [107]; for interpretation, we compare cluster assignments with language families and primary branches.

For two-dimensional visualizations, we reduce the dimensionality of the language vectors using t-SNE [108], UMAP [109], and minimum spanning trees [110]. Although all methods yield valuable insights, we include in the main text only the minimum spanning trees (MST) visualizations colored by language families and subfamilies, as they most clearly represent the inter-language relationships. Additional figures are provided in Appendix C and also available via our open-source tool[3.1].

Datasets and models. In our experiments, we employ three LLMs and three datasets. The models used are Mistral 7B [38], Gemma 3 4B [40], and Llama 3.2 1B [39]. All models are multilingual and have been trained on more than 100 languages. Notably, although Llama officially supports only 8 languages, our results indicate that it still produces useful representations for our purposes. As datasets, we selected those with a high number of languages: Wikipedia [111], CulturaX [112], and fineweb-2 [113].

We start with a target inventory of 106 languages and attempted to apply the same list across all corpora. Wikipedia contains material for every language in this set, but CulturaX omits Chinese (Traditional), Min Nan Chinese, Scots, and Crimean Tatar, whereas fineweb-2 lacks Chinese (Tra-

[3.1] https://huggingface.co/spaces/mshamrai/language-metric-analysis

Dataset	# Languages in Dataset	# Languages Used in Work
Wikipedia	323	106
CulturaX	167	102
fineweb-2	2051	103

Table 3.4. Comparison of datasets: Wikipedia, CulturaX, and fineweb-2. The table reports the total number of languages in each dataset and the number of languages used in this work.

ditional), English[3.2], Serbo-Croatian, and Tagalog. Table 3.4 lists the total number of languages present in each dataset alongside the subset that could be retained from our 106-language list. For the full list of languages see Appendix B.

To compute the language vectors, we proceed as follows:

1. **Calibration data.** For every language in each corpus (Wikipedia, CulturaX, fineweb) we sample $2^{19} = 524{,}288$ tokens.

2. **Weight-importance vectors.** For each language–corpus pair and for each LLM (Mistral 7B, Gemma 3 4B, Llama 3.2 1B) we compute a binary weight importance vector whose length matches the model's parameter count, yielding $3(106 + 102 + 103) = 933$ vectors.

3. **Distance matrices.** Hamming distances between language vectors produce nine language–by–language matrices (one per model–corpus combination).

4. **Aggregation.** These nine matrices are averaged element-wise over the observed entries to form a single average distance matrix.

5. **Embedding.** Classical MDS on the average matrix embeds the languages space in $\mathbb{R}^{104}$, where Euclidean distance defines the final language metric.

[3.2]For the English subset, we use the *fineweb* dataset (https://huggingface.co/datasets/HuggingFaceFW/fineweb)

Evaluation of k-means clustering against two linguistic categorizations. After we embed the $|X| = 106$ language vectors into $\mathbb{R}^{104}$ via classical MDS we evaluate the language embeddings using k-means. The resulting partition is compared with two reference label sets: (i) *high-level families* (18 macro-families) and (ii) *primary branches* (35 sub-families). The number of clusters in k-means is equal to the number of labels in the reference sets.

We compute the following metrics:

- **Silhouette score** [114]: the mean difference between a point's average distance to its own cluster and to the nearest neighboring cluster. Values range from -1 (poor separation) to $+1$ (well-separated, compact clusters).
- **Adjusted Rand Index (ARI)** [115]: agreement between two partitions, corrected for chance. 1 indicates perfect alignment, 0 indicates random overlap.
- **Cluster purity** [116]: the fraction of data points that share the majority label within their cluster. Values in $[0, 1]$.

Reference	Sil.	ARI	Purity
Macro families	0.047	0.116	0.755
Primary branches	0.056	0.434	0.811

Table 3.5. Clustering metrics for the k-means solution against two standard language classification. "Sil" is the internal silhouette score.

Table 3.5 shows that switching from broad families to primary branches raises the ARI from 0.116 to 0.434 and the purity from 0.755 to 0.811. Therefore, the metric space captures finer-grained language groups and can estimate similarity at a micro level. However, the internal silhouette remains low (about 0.05), meaning many languages lie almost as close to other clusters as to their own.

Figure 3.1. Minimum spanning tree for all languages. Colors represent language families.

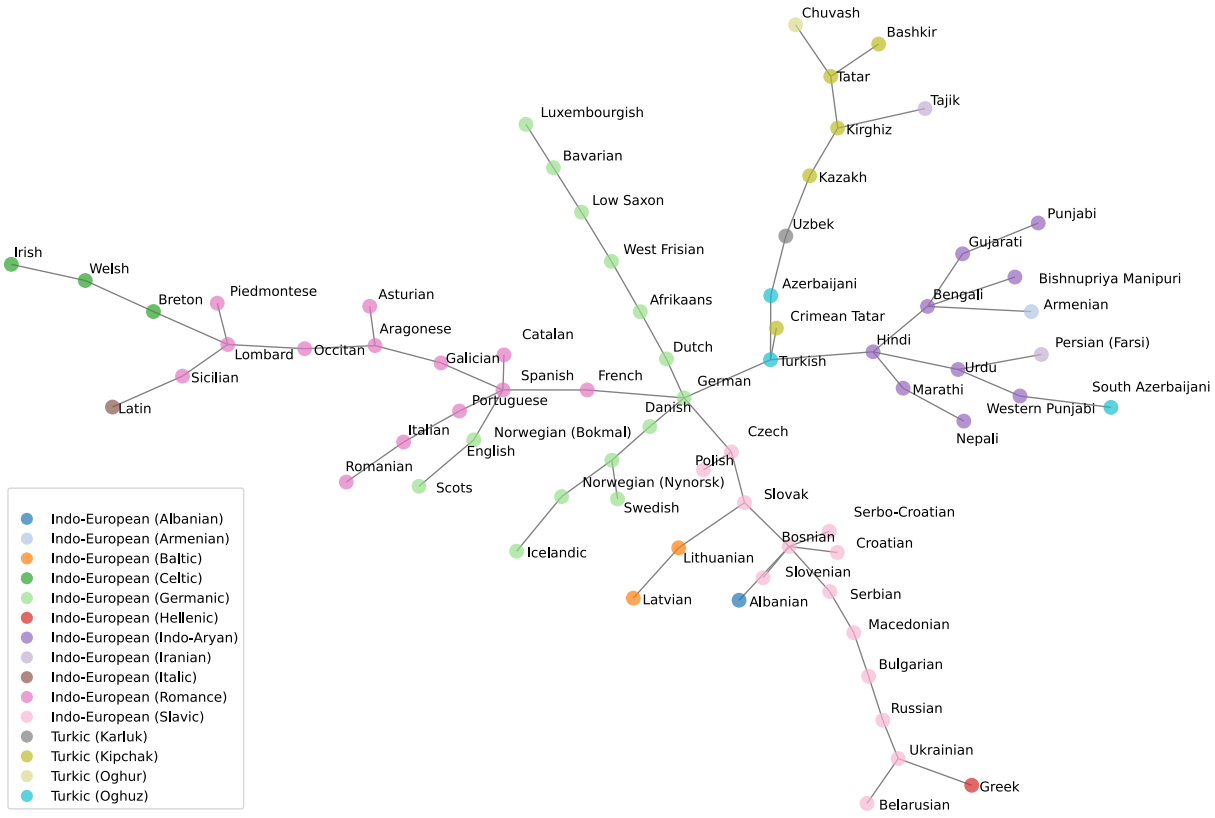


Figure 3.2. Minimum spanning tree for languages from the Indo-European and Turkic families. Colors represent language primary branches.

Language trees. A minimum spanning tree (MST) connects all data points in the dataset with the smallest possible total edge weight, where the edge weight corresponds to the distance between language vectors. We employ the Kamada-Kawai layout, a force-directed algorithm where edge lengths are proportional to the distances [117]. This layout effectively visualizes the structure and connectivity within the MST, revealing not only the clusters of closely related languages but also links between different language families.

Figure 3.1 shows the MST for all languages used in our work. The visualization highlights well-established clusters corresponding to known language families as well as some unexpected connections. For example, Tajik (an Indo-European language) appears linked to a cluster of Turkic languages, which can likely be explained by geographical proximity. Similarly, the branch containing Latvian and Lithuanian is connected to a cluster of Uralic

languages, possibly due to regional contact with Finnish and Estonian. A less obvious connection is observed between Turkish and Hungarian, which might be attributed to historical interactions. Additionally, Vietnamese is found to be close to Chinese, despite Vietnamese using the Latin alphabet and Chinese employing logographic characters, indicating that our method captures internal language characteristics beyond mere orthographic features.

Figure 3.2 focuses on Indo-European and Turkic languages, with coloring based on their primary branches. This figure clearly illustrates that Crimean Tatar, although belonging to the Kipchak branch, is closely connected to Turkish, an Oghuz language. The MST also links English, a Germanic language, directly to Spanish, a Romance language, likely reflecting their close geographic and sociolinguistic contact in the Americas.

One intriguing observation is that Ukrainian does not exhibit a direct connection with Polish in the MST, which is unexpected. However, further analysis reveals that Polish consistently ranks among the top five closest languages to Ukrainian across all models and datasets, coming in third after averaging the distances.

In summary, the minimum spanning trees recover coherent family-level structure while also surfacing atypical cross-family links. These links are plausible candidates for contact phenomena, borrowing, or convergent evolution, and motivate targeted analysis with dedicated linguistic methodology.

Transfer-learning experiments. We investigated whether adding data from a *similar* language can improve a low-resource target, where similarity is measured by the language-distance metric introduced in this paper. All experiments fine-tune Llama 3.2 1B and evaluate exclusively on a held-out set in the target language.

We perform our experiments using the following strategies:

1. **Mixed (size-matched).** An equal amount of auxiliary-language text

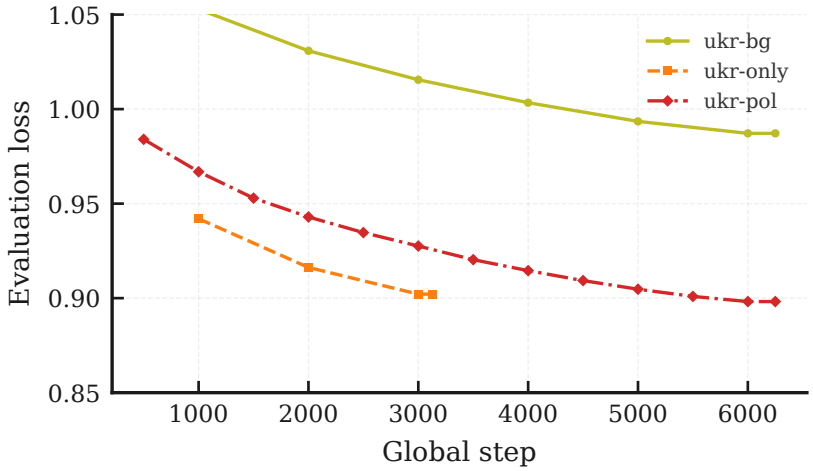


Figure 3.3. Evaluation loss on Ukrainian for three weighted-loss runs: *ukr-only* (baseline), *ukr-bg* (Ukrainian + Bulgarian), and *ukr-pol* (Ukrainian + Polish). Two-language datasets are twice as large, hence the longer training schedule.

is concatenated to the low-resource corpus; the joint data are shuffled and used for fine-tuning.

2. **Mixed (loss-weighted).** The same joint corpus is used, but the loss is re-weighted: e.g. 0.8 for target-language tokens and 0.2 for auxiliary-language tokens.

3. **Sequential.** Fine-tune first on the auxiliary language, then continue training on the low-resource corpus.

Figure 3.3 shows that augmenting Ukrainian with the metrically close Bulgarian does not improve evaluation loss, and Polish yields only a minor reduction.

A similar pattern emerges for sequential fine-tuning on Turkish followed by Crimean Tatar: perplexity drops from 5.48 (Crimean Tatar only) to 5.36, an insignificant change.

Across all settings, none of the three transfer regimes produced a consistent, significant gain over single-language fine-tuning. Future work should revisit these transfer strategies with substantially larger models and much larger datasets, where the benefits of distance-based language pairing may emerge more clearly.

3.3. Closed-form robustness bounds for second-order pruning of neural controller policies

Modern autonomous systems, from agile quadcopters to embodied household robots, are increasingly controlled by *neural policies* that map high-dimensional sensor data directly to control actions. Rich function classes such as multilayer perceptrons (MLPs), convolutional networks, or *Vision–Language–Action* (VLA) models achieve impressive closed-loop performance when trained by reinforcement learning (RL) or behaviour cloning [24, 25]. Yet deploying those policies on size, weight, and power-constrained hardware demands aggressive *model compression*. Among the most practical compression techniques are *second-order pruning* algorithms: Optimal Brain Damage (OBD) [71], Optimal Brain Surgeon (OBS) [72], and their recent large-model successor *SparseGPT* [7], which remove weights that minimise an analytically estimated activation loss.

While second-order pruning is widely used for large language models, its impact on the *closed-loop behaviour* of a control system is poorly investigated. A small perturbation of the weights can propagate through the policy network, alter the control signal, and ultimately degrade safety or task performance. Precise guarantees on how much pruning a controller can tolerate are therefore crucial for safety-critical applications [30–33]. To the best of our knowledge, *no prior work provides a rigorous, closed–form upper bound on the control error induced by second–order pruning* in nonlinear systems.

We provide the first *robustness analysis* of OBD/OBS-style weight pruning for deterministic nonlinear control systems. Our goal is to bound, in closed form, the deviation of the pruned control signal.

Contributions.

1. **Formal robustness framework.** Subsection 3.3.1 casts pruning as a perturbation of the parameter vector Θ and formulates the *pruning robustness problem* via the control–error bound (3.7).

2. **Tight single–layer bound.** Theorem 3.10 proves that for any 1–Lipschitz (ReLU–type) activation the deviation of a pruned layer k is

$$\left\|\pi(s;\Theta) - \pi(s;\widehat{\Theta})\right\|_2 \;\le\; C_k(s)\,\|\delta W_k\|_2,$$

where the constant $C_k(s)$ depends only on *unpruned* weights, biases and the input norm.

3. **Extension to multiple layers.** Corollary 3.11 extends the result to an arbitrary set of pruned layers in an *additive* fashion, yielding the computable bound $B_\pi(\delta\Theta) = \sum_{k\in S} C_{k,\max}\|\delta W_k\|_2$.

The bounds can be evaluated *offline* from a forward pass and spectral norms alone, enabling principled trade–offs between compression ratio and control robustness without repeated interaction with the physical system.

3.3.1. Formal problem setup.

Deterministic nonlinear control problem.

Definition 3.1 (Controlled dynamics)**.** Let the **state space** be $X \subset \mathbb{R}^n$ and the **action space** be $U \subset \mathbb{R}^m$. A trajectory $\{x_t\}_{t\ge 0}$ evolves according to the discrete–time difference equation

$$x_{t+1} \;=\; f\big(x_t, u_t\big), \qquad t = 0, 1, 2, \ldots,$$

where the transition map $f : X \times U \to X$ is continuous, ensuring that a unique trajectory exists for every admissible control sequence $\{u_t\}_{t\ge 0}$ and initial state $x_0 \in X$.

Definition 3.2 (Parametric neural policy)**.** Let $L \in \mathbb{N}$ be the number of affine layers. Collect all weights and biases in the parameter vector

$$\Theta \;=\; \big\{W_\ell, b_\ell\big\}_{\ell=1}^{L} \in \mathbb{R}^q.$$

The **policy** $\pi(\,\cdot\,;\Theta) : X \to U$ is the neural network obtained by composing these affine layers with pointwise, 1–Lipschitz activations σ:

$$\pi(x;\Theta) \;:=\; (\sigma\circ A_L)\circ\cdots\circ(\sigma\circ A_1)(x), \qquad A_\ell(z) \;=\; W_\ell z + b_\ell.$$

The control applied at time t is $u_t = \pi(x_t;\Theta)$.

Definition 3.3 (Discounted return)**.** Fix an initial distribution $x_0 \sim \mu$, a bounded reward $r : X \times U \to \mathbb{R}$, and a discount factor $\gamma \in (0,1)$. The expected return of policy $\pi(\cdot;\Theta)$ is

$$J(\Theta) \;:=\; \mathbb{E}_{x_0\sim\mu}\big[\textstyle\sum_{t=0}^{\infty}\gamma^t\, r\big(x_t,\pi(x_t;\Theta)\big)\big].$$

An **optimal policy** is any maximiser $\Theta^\star \in \arg\max_{\Theta\in\mathbb{R}^q} J(\Theta)$.

Second–order (OBD) pruning criterion. Throughout this paper we study the effect of pruning individual weights in Θ by the *Optimal Brain Damage (OBD)* saliency [71]. Consider a single affine layer $A_\ell(z) = W_\ell z + b_\ell$ and an input mini–batch $X_\ell \in \mathbb{R}^{d\times n_\ell}$. The change in pre–activation outputs caused by replacing W_ℓ with $\widehat{W}_\ell = W_\ell + \delta W_\ell$ is measured by

$$E_\ell(W_\ell,\widehat{W}_\ell) \;=\; \big\|W_\ell X_\ell - \widehat{W}_\ell X_\ell\big\|_F^2. \tag{3.5}$$

A second–order Taylor expansion of E_ℓ around W_ℓ yields

$$E_\ell(W_\ell+\delta W_\ell) \;\approx\; \tfrac{1}{2}\,\mathrm{vec}(\delta W_\ell)^\top H_\ell\;\mathrm{vec}(\delta W_\ell), \qquad H_\ell := \nabla^2_{W_\ell} E_\ell\,.$$

Pruning a *single* weight $w_q \in \Theta$ (that is, setting $\widehat{w}_q = 0$) gives the OBD saliency

$$\Delta E_q \;=\; \frac{1}{2}\,\frac{w_q^2}{(H_\ell^{-1})_{qq}}, \tag{3.6}$$

which serves as an importance score for that weight. Aggregating (3.6) over the layers to which OBD is applied produces a pruned parameter vector $\widehat{\Theta}\in\mathbb{R}^q$ and the perturbation $\delta\Theta := \widehat{\Theta} - \Theta$.

Pruning robustness problem. Our objective is to quantify **how sensitive the closed–loop behaviour is to the OBD perturbation** $\delta\Theta$. Specifically, we seek a computable bound $B_\pi : \mathbb{R}^q \to \mathbb{R}_+$ such that

$$\left\|\pi(x;\Theta) - \pi(x;\widehat{\Theta})\right\|_2 \;\le\; B_\pi(\delta\Theta), \qquad \forall\, x \in X. \tag{3.7}$$

Although inequality (3.7) applies to *any* feed–forward network, we focus on the control setting because the resulting guarantees translate directly into performance and safety margins for autonomous systems.

Remark 3.4. For clarity, we treat each weight matrix W_ℓ and bias b_ℓ as a distinct block within Θ. Hence any scalar weight w_q appearing in (3.6) is an element of Θ, and replacing w_q by 0 modifies Θ exactly as required in the control bound (3.7).

3.3.2. Background.

All robustness estimates in this section rest on *Lipschitz control* of the policy network. We collect the required analytic facts in this section.

Lipschitz mappings.

Definition 3.5 (Global Lipschitz continuity)**.** A function $f : \mathbb{R}^n \to \mathbb{R}^m$ is called L–**Lipschitz** with respect to the Euclidean norm if

$$\|f(x) - f(y)\|_2 \;\le\; L\,\|x - y\|_2, \qquad \forall\, x, y \in \mathbb{R}^n. \tag{3.8}$$

The smallest such constant is denoted $L(f)$.

Theorem 3.6 (Rademacher [118])**.** *Every locally Lipschitz map $f : \mathbb{R}^n \to \mathbb{R}^m$ is differentiable almost everywhere and $L(f) \;=\; \operatorname{ess\,sup}_x \|Df(x)\|_2$, where $\|\cdot\|_2$ is the operator norm induced by ℓ_2.*

3.3.3. Lipschitz multilayer perceptrons.

Definition 3.7 (MLP with 1–Lipschitz activations)**.** Fix $L \in \mathbb{N}$. Let $\sigma : \mathbb{R} \to \mathbb{R}$ satisfy $|\sigma(a) - \sigma(b)| \le |a - b|$. An ***L*-layer MLP** is the map

$$f(x; \Theta) := (\sigma \circ A_L) \circ \cdots \circ (\sigma \circ A_1)(x), \qquad A_\ell(z) = W_\ell z + b_\ell,$$

with parameters $\Theta = \{W_\ell, b_\ell\}_{\ell=1}^{L}$ and $W_\ell \in \mathbb{R}^{d_\ell \times d_{\ell-1}}$.

Proposition 3.8 (Spectral-norm Lipschitz bound [119])**.** *For the MLP of Definition 3.7,*

$$L\big(f(\,\cdot\,;\Theta)\big) \;\le\; \prod_{\ell=1}^{L} \|W_\ell\|_2. \tag{3.9}$$

Interpretation for pruning analysis.. Equation (3.9) expresses the *global* Lipschitz constant of a neural policy directly in terms of the spectral (operator-norm) factors that appear in the OBD parameter vector Θ. Hence perturbing any weight matrix $W_k \mapsto W_k + \delta W_k$ alters both $L\big(f(\cdot;\Theta)\big)$ and the saliency (3.6), allowing us to translate the activation-level OBD error into a control-signal bound (3.7) in later sections.

3.3.4. Results.

Proposition 3.9 (Non-expansiveness of ReLU-type activations)**.** *Let $\varphi : \mathbb{R} \to \mathbb{R}$ be an activation satisfying*

1. zero anchor: $\varphi(0) = 0$,
2. unit Lipschitz: $|\varphi(a) - \varphi(b)| \le |a - b|$ *for all* $a, b \in \mathbb{R}$.

Define the component-wise map

$$\sigma : \mathbb{R}^n \to \mathbb{R}^n, \qquad \sigma(x)_i \;=\; \varphi(x_i),\; i = 1, \ldots, n.$$

*Then σ is ℓ_2–*non-expansive*:*

$$\|\sigma(x)\|_2 \;\le\; \|x\|_2, \qquad \forall\, x \in \mathbb{R}^n. \tag{3.10}$$

Proof. For any $x \in \mathbb{R}^n$ we have, by the scalar Lipschitz property with $b = 0$, $|\varphi(x_i)| \le |x_i|$. Squaring, summing and taking the square root yields

$$\|\sigma(x)\|_2^2 \;=\; \sum_{i=1}^{n} \varphi(x_i)^2 \;\le\; \sum_{i=1}^{n} x_i^2 \;=\; \|x\|_2^2,$$

from which (3.10) follows. □

Condition (3.10) is satisfied by many ReLU-type activations that are ubiquitous in deep learning:

- **ReLU** $\varphi(x) = \max\{0, x\}$ [14];
- **Leaky-ReLU** $\varphi(x) = \max\{x, \alpha x\}$ with $0 < \alpha \le 1$ [16];
- **PReLU** (parametric ReLU) with learnable $\alpha \in (0, 1]$ [15];
- **ELU** $\varphi(x) = \max\{x, \alpha(e^x - 1)\}$ for $0 < \alpha \le 1$ [17];
- **GELU** (Gaussian-error linear unit), a smooth approximation to ReLU whose derivative is bounded by 1 [18].

All these functions obey $\varphi(0) = 0$ and have slope bounded by 1, hence are 1-Lipschitz, therefore Proposition 3.9 applies.

Theorem 3.10 (Robustness of an OBD-pruned policy)**.** *Let $\pi(\,\cdot\,;\Theta)$ be the L-layer MLP controller*

$$x_0 = s, \qquad x_\ell = \sigma\big(W_\ell x_{\ell-1} + b_\ell\big),\ \ell = 1, \dots, L, \quad \pi(s;\Theta) = x_L,$$

with ReLU-type activations σ_l and weights $\Theta = \{W_\ell, b_\ell\}_{\ell=1}^{L}$. Suppose layer k is pruned, giving $\widehat{W}_k = W_k + \delta W_k$ and $\widehat{\Theta} = \Theta + \delta\Theta$. Then, for every input state $s \in X$,

$$\big\|\pi(s;\Theta) - \pi(s;\widehat{\Theta})\big\|_2 \;\le \tag{3.11}$$

$$\le\; \|\delta W_k\|_2 \Big(\|s\|_2 \prod_{\substack{\ell=1 \\ \ell \ne k}}^{L} \|W_\ell\|_2 + \sum_{i=1}^{k-1} \prod_{\substack{\ell=i+1 \\ \ell \ne k}}^{L} \|W_\ell\|_2 \|b_i\|_2 \Big) \;\le \tag{3.12}$$

$$\leq \|\delta W_k\|_2 \underbrace{\Big(\sup_{s\in X} \|s\|_2 \prod_{\substack{\ell=1 \\ \ell\neq k}}^{L} \|W_\ell\|_2 + \sum_{i=1}^{k-1} \prod_{\substack{\ell=i+1 \\ \ell\neq k}}^{L} \|W_\ell\|_2 \|b_i\|_2 \Big)}_{=:\, C_{\max}}. \tag{3.13}$$

Consequently, the control-error bound in (3.7) can be chosen as $B_\pi(\delta\Theta) = C_{\max}\|\delta W_k\|_2$.

Proof. Because σ_l are 1-Lipschitz, $\|\sigma_l(u) - \sigma_l(v)\|_2 \leq \|u - v\|_2$. Write $\delta x_\ell := x_\ell - \widehat{x}_\ell$, then for the last layer

$$\|\pi(s;\Theta) - \pi(s;\widehat{\Theta})\|_2 = \|x_L - \widehat{x}_L\|_2 = \|\delta x_L\|_2,$$

and

$$\begin{aligned} \|\delta x_L\|_2 &= \|\sigma_L(W_L x_{L-1} + b_L) - \sigma_L(W_L \widehat{x}_{L-1} + b_L)\|_2 \\ &\leq \|W_L x_{L-1} - W_L \widehat{x}_{L-1}\|_2 \leq \|W_L\|_2 \|\delta x_{L-1}\|_2 \\ &\Rightarrow \|\delta x_L\|_2 \leq \|W_L\|_2 \|\delta x_{L-1}\|_2. \end{aligned}$$

Iterating the argument down to layer k yields

$$\begin{aligned} \|\delta x_L\|_2 &\leq \Big(\prod_{\ell=k+1}^{L} \|W_\ell\|_2 \Big) \|W_k x_{k-1} - \widehat{W}_k x_{k-1}\|_2 \\ &= \Big(\prod_{\ell=k+1}^{L} \|W_\ell\|_2 \Big) \|\delta W_k x_{k-1}\|_2 \\ &\leq \Big(\prod_{\ell=k+1}^{L} \|W_\ell\|_2 \Big) \|\delta W_k\|_2 \|x_{k-1}\|_2. \end{aligned}$$

Because σ_l are ReLU-type by Proposition 3.9

$$\begin{aligned} \|x_{k-1}\|_2 &= \|\sigma_{k-1}(W_{k-1} x_{k-2} + b_{k-1})\|_2 \\ &\leq \|W_{k-1} x_{k-2} + b_{k-1}\|_2 \\ &\leq \|W_{k-1}\|_2 \|x_{k-2}\|_2 + \|b_{k-1}\|_2 \\ &\leq \cdots \leq \end{aligned}$$

$$\leq \|s\|_2 \prod_{\ell=1}^{k-1} \|W_\ell\|_2 + \sum_{i=1}^{k-2} \prod_{\ell=i+1}^{k-1} \|W_\ell\|_2 \|b_i\|_2 + \|b_{k-1}\|_2.$$

Substituting gives

$$\begin{aligned}
&\|\pi(s;\Theta) - \pi(s;\widehat{\Theta})\|_2 \leq \\
&\leq \Big(\prod_{\ell=k+1}^{L} \|W_\ell\|_2 \Big) \|\delta W_k\|_2 \Big(\|s\|_2 \prod_{\ell=1}^{k-1} \|W_\ell\|_2 + \\
&+ \sum_{i=1}^{k-2} \prod_{\ell=i+1}^{k-1} \|W_\ell\|_2 \|b_i\|_2 + \|b_{k-1}\|_2 \Big) \leq \\
&\leq \|\delta W_k\|_2 \Big(\|s\|_2 \prod_{\substack{\ell=1 \\ \ell \neq k}}^{L} \|W_\ell\|_2 + \sum_{i=1}^{k-1} \prod_{\substack{\ell=i+1 \\ \ell \neq k}}^{L} \|W_\ell\|_2 \|b_i\|_2 \Big),
\end{aligned}$$

which completes the bound 3.11. □

Corollary 3.11 (Robustness under pruning *multiple* layers)**.** *Let the assumptions of Theorem 3.10 hold and let* $S = \{k_1, \ldots, k_m\} \subseteq \{1, \ldots, L\}$ *be an index set of layers pruned by OBD. For every* $k \in S$ *write* $\widehat{W}_k = W_k + \delta W_k$ *and* $\widehat{\Theta} = \Theta + \delta\Theta$ *with* $\delta\Theta = \{\delta W_k\}_{k \in S}$*. Define, for each* $k \in S$ *and input state* $s \in X$*,*

$$C_k(s) \;:=\; \|s\|_2 \prod_{\ell \neq k} \|W_\ell\|_2 + \sum_{i=1}^{k-1} \Big(\prod_{\substack{\ell=i+1 \\ \ell \neq k}}^{L} \|W_\ell\|_2 \Big) \|b_i\|_2,$$

$$C_{k,\max} := \sup_{s \in X} C_k(s).$$

Then, for every $s \in X$*,*

$$\big\|\pi(s;\Theta) - \pi(s;\widehat{\Theta})\big\|_2 \;\leq\; \sum_{k \in S} \|\delta W_k\|_2 \, C_k(s) \tag{3.14}$$

$$\leq \sum_{k \in S} \|\delta W_k\|_2 \, C_{k,\max}. \tag{3.15}$$

Consequently, a valid control-error budget in (3.7) is $B_\pi(\delta\Theta) = \sum_{k\in S} C_{k,\max}\,\|\delta W_k\|_2$.

Proof. Order the pruned layers as $k_1 < \cdots < k_m$ and construct an intermediate parameter sequence $\Theta^0 := \Theta,\ \Theta^j := \Theta^{j-1} + \delta W_{k_j},\ j = 1, \ldots, m$, so that $\Theta^m = \widehat{\Theta}$.

Applying Theorem 3.10 to the pair (Θ^{j-1}, Θ^j) and layer k_j gives

$$\left\|\pi(s;\Theta^{j-1}) - \pi(s;\Theta^j)\right\|_2 \;\le\; \|\delta W_{k_j}\|_2\, C_{k_j}(s).$$

Summing these m inequalities and using the triangle inequality yields the first bound in (3.14). Taking the supremum over s establishes the second bound. □

Corollary 3.11 shows that control robustness degrades *additively* with respect to the spectral-norm perturbations $\{\delta W_k\}_{k\in S}$. The constants $C_k(s)$ depend only on the unpruned weights, biases, and the input magnitude, making them computable *before* pruning takes place. In practice one may evaluate the tighter, state-dependent bound $C_k(s)$ on a validation set or deploy the uniform constant $C_{k,\max}$ for worst-case guarantees.

3.4. Nonasymptotic bounds on return degradation for OBD-pruned neural controllers

This section shares the same object of study and perturbation model as Section 3.3: second-order-pruned neural policies. To avoid repeating motivation and compression background, we focus directly on the return-level question.

Concretely, we lift the output-space robustness analysis of Section 3.3 to discounted-return guarantees by combining: (i) performance-difference bounds in total variation (Theorem 3.19); (ii) a second-order OBD link that yields an explicit return certificate (Corollary 3.21); and (iii) a scalable layer-local bound that avoids global Hessian estimation (Corollary 3.22).

3.4.1. Preliminaries. Unless noted otherwise, we reuse the policy/pruning notation and assumptions from subsection 3.3.1. This subsection introduces only the RL-specific objects required for return-level analysis.

Fundamental notations are: *discount factor:* $\gamma \in (0,1)$ controls the importance of future rewards; $\gamma \approx 1$ encourages long-term planning, whereas small γ focuses on immediate gains; *state & action spaces:* $\mathcal{S}$ and $\mathcal{A}$ are *finite* sets of states and actions; *dynamics and rewards:* $P(\, s' \mid s, a)$ is the state-transition kernel and the reward is bounded: $0 \le r(s,a) \le R_{\max} < \infty$ for all (s,a); *norms:* bold symbols (e.g. $\mathbf{x}$) denote vectors, $\|\cdot\|_1$ and $\|\cdot\|_2$ are the usual ℓ_1 (sum) and ℓ_2 (Euclidean) norms; *probability simplices:* $\Delta(\mathcal{A})$ denotes the set of all distributions over $\mathcal{A}$; *we treat probability distributions as column vectors whose entries sum to* 1 (both norm and linear algebra notation apply seamlessly).

Definition 3.12 (Markov Decision Process)**.** A *Markov Decision Process* (MDP) is the tuple $\langle \mathcal{S}, \mathcal{A}, P, r, \gamma, \rho \rangle$, where ρ is the distribution of the initial state s_0.

Definition 3.13 (Policy)**.** A *policy* is any measurable mapping $\pi : \mathcal{S} \to \Delta(\mathcal{A})$, $s \mapsto \pi(\cdot \,|\, s)$. We write $a \sim \pi(\cdot \,|\, s)$ when sampling an action from π in state s.

A *pruned policy* π' is obtained from a given neural policy π_θ by setting a subset of network parameters to zero using *Optimal Brain Damage* (OBD) [71]. We use $\pi' = \pi_{\theta'}$ to emphasize the new parameters θ'.

Definition 3.14 (Value, action-value, and advantage)**.** For any policy π and state-action pair (s,a),

$$
\begin{aligned}
V^\pi(s) &:= \mathbb{E}_\pi\Big[\textstyle\sum_{t=0}^{\infty} \gamma^t r(s_t,a_t) \;\Big|\; s_0 = s\Big], \\
Q^\pi(s,a) &:= \mathbb{E}_\pi\Big[\textstyle\sum_{t=0}^{\infty} \gamma^t r(s_t,a_t) \;\Big|\; s_0 = s, a_0 = a\Big], \\
A^\pi(s,a) &:= Q^\pi(s,a) - V^\pi(s).
\end{aligned}
$$

Intuition. $V^\pi(s)$ is the expected *return* starting from s when following π; $Q^\pi(s,a)$ is the same, assuming we first force action a. The *advantage* A^π

measures how much better (or worse) a is compared to the average prescribed by π in s.

Definition 3.15 (Discounted visitation distribution)**.** The *discounted state visitation* under π is

$$d^{\pi}(s) \;:=\; (1-\gamma)\sum_{t=0}^{\infty}\gamma^{t}\,\Pr(s_t = s \mid \pi, \rho),$$

which places more weight on states visited earlier in an episode.

Definition 3.16 (Expected return (performance))**.** Given an initial-state distribution ρ, the (discounted) expected return of policy π is

$$J(\pi) \;:=\; \mathbb{E}_{s_0\sim\rho}\big[V^{\pi}(s_0)\big] \;=\; \frac{1}{1-\gamma}\,\mathbb{E}_{s\sim d^{\pi},\,a\sim\pi(\cdot|s)}\big[r(s,a)\big],$$

where the second equality is a standard identity obtained by unrolling the definition of d^{π} in Definition 3.15. For a pruned policy π', we write $J(\pi')$ analogously.

Definition 3.17 (Total variation and KL divergence)**.** For distributions $p, q \in \Delta(\mathcal{A})$,

$$D_{\mathrm{TV}}(p,q) := \tfrac{1}{2}\sum_{a\in\mathcal{A}}|p(a)-q(a)|, \qquad D_{\mathrm{KL}}(p\|q) := \sum_{a\in\mathcal{A}}p(a)\log\frac{p(a)}{q(a)}.$$

Lemma 3.18 (Performance–difference lemma [120])**.** *For any two policies* π, π',

$$J(\pi') - J(\pi) \;=\; \frac{1}{1-\gamma}\,\mathbb{E}_{s\sim d^{\pi'}}\mathbb{E}_{a\sim\pi'}\big[A^{\pi}(s,a)\big].$$

We parameterize the policy by $\theta \in \mathbb{R}^d$ and write π_θ. *Pruning* removes parameters to reduce memory and inference cost. Formally, let $m \in \{0,1\}^d$ be a binary *mask* and define the pruned parameters $\theta' := m \odot \theta$ (Hadamard product). The parameter perturbation is $\delta := \theta' - \theta = (m - \mathbf{1}) \odot \theta$, so that $\delta_i = -\theta_i$ for pruned indices and 0 otherwise. For layered networks we also use layerwise notation $\{W_\ell, b_\ell\}_{\ell=1}^{L}$ and masks M_ℓ so that $\widehat{W}_\ell := M_\ell \odot W_\ell$.

Optimal Brain Damage (OBD) selects weights to prune by minimizing a second-order approximation to the post-pruning loss. If $h_i \geq 0$ denotes the ith diagonal entry of the Hessian of the loss (or a suitable layer-local surrogate), the *saliency* of weight θ_i is $s_i := h_i \theta_i^2$. Under a global sparsity budget $\tau \in (0, 1)$ (or a target of k weights), OBD prunes the indices with the *smallest* s_i, yielding θ' that (approximately) minimizes the predicted loss increase subject to the budget. In the sequel we relate this parameter change δ (or layerwise $\|\delta W_k\|_2$) to changes in the *behavior* of the policy, quantified by $D_{\mathrm{TV}}\big(\pi'(\cdot \mid s), \pi(\cdot \mid s)\big)$ and ultimately by the return gap $|J(\pi') - J(\pi)|$.

To connect small parameter perturbations to changes in the controller's outputs, we directly invoke the deterministic single-layer robustness theorem established in the previous section (Theorem 3.10). In particular, the bound in equation (3.11) gives $\|\pi(s; \Theta) - \pi(s; \widehat{\Theta})\|_2 \leq C_{\max} \|\delta W_k\|_2$, which we use below to convert policy perturbation into a return certificate.

3.4.2. Results.

Theorem 3.19 (Return difference via total variation)**.** *In any finite MDP with $r \in [0, R_{\max}]$ and discount γ,*

$$|J(\pi') - J(\pi)| \;\leq\; \frac{2R_{\max}}{(1-\gamma)^2}\, \mathbb{E}_{s \sim d^{\pi'}} \Big[D_{\mathrm{TV}}\big(\pi'(\cdot \,|\, s),\, \pi(\cdot \,|\, s)\big) \Big].$$

Corollary 3.20 (KL–based version)**.** *Combining Theorem 3.19 with Pinsker's inequality gives*

$$|J(\pi') - J(\pi)| \;\leq\; \frac{2R_{\max}}{(1-\gamma)^2}\, \mathbb{E}_{s \sim d^{\pi'}} \Big[\sqrt{\tfrac{1}{2} D_{\mathrm{KL}}\big(\pi' \| \pi\big)} \Big].$$

A simpler (conservative) worst-case bound is obtained by replacing the expectation with a supremum over s.

Corollary 3.21 (OBD performance bound)**.** *Let $\pi_{\theta^\star}$ be a locally optimal network policy with vanishing first derivatives,*
$\nabla_\theta \pi_{\theta^\star}(a \,|\, s) = 0$. *Prune weights $\mathcal{P} \subset [d]$ to obtain $\theta' = \theta^\star + \delta$, $\delta_i = -\theta_i^\star$ for*

$i \in \mathcal{P}$. *If each* $\theta \mapsto \pi_\theta(a\,|\,s)$ *is* $\mathcal{C}^3$ *with bounded third derivatives, then*

$$|J(\pi_{\theta'}) - J(\pi_{\theta^\star})| \;\le\; \frac{R_{\max}}{2(1-\gamma)^2}\, S_{\mathrm{OBD}} + \mathcal{O}(\|\delta\|_2^3),$$

where $S_{\mathrm{OBD}} := \sum_{i\in\mathcal{P}} \Big[\sum_{(s,a)} h_i^{(a,s)}\Big] \theta_i^{\star\,2}$ *is the sum of* diagonal Hessian scores *used by OBD [71].*

Intuition. S_{OBD} is precisely what OBD minimises; the corollary states that, for small enough pruning, the drop in return is *linear* in that score.

The diagonal entries $h_i^{(a,s)}$ in Corollary 3.21 come from the *global* Hessian $H_{a,s} \in \mathbb{R}^{d\times d}$, where $d \approx 10^8$ for contemporary vision or language policies. Even storing $H_{a,s}$ is infeasible, let alone computing it for every (s,a). SparseGPT [7] circumvents this by optimising a simple quadratic objective $\frac{1}{2}\|XW - Y\|_F^2$ *layer by layer*, whose Hessian $X^\top X$ is small and easy to invert. Thus curvature is captured *locally* while memory scales linearly in the number of parameters.

Corollary 3.22 (Return bound via layer–local TV (single pruned layer))**.** *Assume the setting of Theorem 3.10 and that the policy head mapping the final network output to action probabilities is* 1*–Lipschitz in* ℓ_2 *(e.g., a softmax over logits) so that* $\pi(\cdot\mid s), \pi'(\cdot\mid s) \in \Delta(\mathcal{A})$ *with* $A := |\mathcal{A}| < \infty$*. Then*

$$\big|J(\pi') - J(\pi)\big| \;\le\; \frac{R_{\max}\sqrt{A}\, C_{\max}}{(1-\gamma)^2}\, \big\|\delta W_k\big\|_2.$$

The corollary leverages the layer–wise output perturbation control from Theorem 3.10 to upper bound the *statewise* policy shift in total variation, and then propagates this shift to a guaranteed return bound via Theorem 3.19. The certificate is: (i) *linear* in the pruning magnitude $\|\delta W_k\|_2$; (ii) *dimension–free* with respect to the total parameter count; and (iii) dependent on the action–space size only through $\sqrt{A}$. A less conservative, distributional variant replaces $\sup_s$ by $\mathbb{E}_{s\sim d^{\pi'}}[\cdot]$, yielding the same scaling but with $C_{\max}$ multiplied by an average instead of a worst–case factor. For multiple pruned layers, a triangle–inequality extension gives $\|\pi' - \pi\|_2 \le \sum_{k\in\mathcal{K}} C_{\max}^{(k)} \|\delta W_k\|_2$, leading to an additive bound on the return gap.

3.5. Conclusion

This chapter established a unified perturbation perspective on pruning, in which parameter changes $\delta\Theta$ are systematically mapped to task-level effects. The chapter contributes both empirical evidence (language modeling and language geometry) and theoretical guarantees (control robustness and RL return degradation).

Integrated findings.

1. **Language-specific LLM pruning.** Under matched sparsity budgets, SparseGPT consistently outperforms Wanda in quality retention, especially for 2:4 semi-structured pruning. Calibration-language mismatch (English calibration, Ukrainian evaluation) degrades performance, confirming that pruning saliency is language-dependent.

2. **Metric-space construction from pruning saliency.** Binary weight-importance vectors induce a meaningful language geometry that aligns with established family structure while revealing plausible cross-family contacts. The framework is scalable, model-agnostic, and reproducible.

3. **Certified behavior degradation in control and RL.** For deterministic nonlinear control, Theorem 3.10 and Corollary 3.11 provide closed-form layer-local robustness bounds. For RL, Theorem 3.19, Corollary 3.21, and Corollary 3.22 yield a practical pipeline

$$\|\delta W_k\|_2 \;\Rightarrow\; D_{\mathrm{TV}}(\pi', \pi) \;\Rightarrow\; |J(\pi') - J(\pi)|,$$

enabling pre-pruning budgeting and post-pruning validation without global Hessian estimation.

Limitations and future work.

- **Computational cost of geometry extraction.** Constructing binary language vectors remains expensive (e.g., approximately 20 minutes per vector on Mistral 7B with RTX 3090).
- **Model scale and bias.** Experiments have not yet been extended to substantially larger models; averaging across several LLMs reduces but does not eliminate shared inductive biases, especially for low-resource languages.
- **Transfer-learning validation gap.** Preliminary distance-guided transfer experiments did not produce statistically significant improvements, indicating that language distance alone is insufficient as a transfer policy.
- **Conservativeness of theoretical bounds.** The guarantees rely on deterministic dynamics, 1-Lipschitz activations, and worst-case spectral-norm constants; they do not yet certify long-horizon safety constraints under stochastic disturbances.

The next stage of this research is therefore threefold: (i) extend empirical studies to larger and more diverse LLMs, (ii) identify the subset of layers that dominates language-geometry signals, and (iii) tighten return- and safety-aware pruning certificates by incorporating distributional and environment-specific structure.

Conclusion

This dissertation develops a unified perturbation-based framework for model compression and demonstrates how theoretical guarantees can be translated into practical compression decisions. The work combines matrix perturbation theory, low-rank approximation, language-model pruning, and control/RL robustness analysis. The central thesis is that compression quality should be evaluated not only by parameter reduction, but by explicit control of how parameter perturbations propagate to spectral structure, language quality, and decision-making performance.

The main results can be summarized as follows.

1. **Further developed perturbation analysis for concatenated matrices.** In Section 2.1, we derived explicit upper bounds for singular-value deviations under blockwise perturbations and compared two Gram-matrix viewpoints, $M^\top M$ and $MM^\top$. The analysis clarifies when each formulation yields a tighter estimate and shows that the $MM^\top$-based bound is often more informative in practical settings. These results provide a quantitative criterion for monitoring spectral stability when multiple matrices are jointly compressed.

2. **First obtained a compression-aware clustering with explicit reconstruction control.** In Section 2.2, we formulated grouping of matrices as a constrained feasibility problem: a candidate group is accepted only if the predicted truncated-SVD reconstruction error satisfies a prescribed tolerance. This replaces heuristic grouping rules with a principled merge certificate tied directly to low-rank reconstruction quality.

3. **Further developed algorithms for scalable concatenated SVD compression.** Also, in Section 2.2, we developed three complementary strategies for matrix grouping and compression: max-norm (fast and conservative), residual-based (most reliable), and approximate incremental (scalable). Their analysis and experiments demonstrate a consistent computational trade-off between speed, tightness of guarantees, and achievable compression.

4. **Improved empirical analysis of pruning for multilingual LLMs.** In Section 3.1, we studied training-free pruning under fixed sparsity and calibration regimes. The experiments show that SparseGPT is generally more stable than Wanda in quality retention, especially under 2:4 semi-structured pruning, and that calibration-language mismatch degrades downstream quality, confirming language-dependent pruning saliency.

5. **First obtained a pruning-based language metric space.** In Section 3.2, we proposed a representation of language geometry based on binary weight-importance vectors and applied it to 106 languages. Distances in this space recover meaningful family-level structure and reveal cross-family interactions, demonstrating that pruning signals can serve as a quantitative source for comparative linguistic analysis.

6. **Further developed closed-form robustness guarantees for pruned neural controllers.** In Section 3.3, we derived explicit control-error bounds for OBD/OBS-style pruning of multilayer neural policies with 1-Lipschitz activations. The bounds are computable from unpruned quantities, extend additively across layers, and provide a practical robustness budget.

7. **Further developed nonasymptotic return-degradation bounds in reinforcement learning.** In Section 3.4, we linked parameter perturbations to policy divergence in total variation and then to return loss

guarantees, establishing a certification chain

$$\|\delta W_k\|_2 \Rightarrow D_{\mathrm{TV}}(\pi', \pi) \Rightarrow |J(\pi') - J(\pi)|,$$

including an OBD-based bound and a layer-local variant that avoids global Hessian computation.

From a practical perspective, the dissertation provides tools for three deployment scenarios: (i) error-controlled low-rank compression of matrix collections, (ii) language-aware pruning of large language models, and (iii) safety-oriented pruning of neural controllers and RL policies. Across all three scenarios, the same methodological principle is used: design compression decisions around perturbation bounds that are interpretable, computable, and directly connected to application-level objectives.

The future work can be summarized as:

1. **Joint optimization of rank and clustering.** Develop methods where the target rank r is selected adaptively during merging, rather than fixed in advance, to address the observed non-monotonic dependence of compression on rank.

2. **Online and streaming compression-aware clustering.** Extend the offline framework to sequential settings where new matrices are incorporated incrementally using efficient per-cluster error estimation and dynamic cluster management.

3. **Alternative reconstruction norms with certified merge rules.** Generalize beyond Frobenius-error criteria (e.g., toward spectral-norm control) by designing new certificates that directly control critical singular-value tails of concatenated matrices.

4. **Task-aware clustering objectives.** Incorporate downstream-task signals into compression-driven clustering so that grouping decisions preserve task performance, not only reconstruction fidelity.

5. **Randomized/sketch-based acceleration with safety guarantees.** Integrate randomized SVD and sketching ideas to reduce computational cost while preserving conservative, certificate-based merge decisions under explicit error tolerances.

6. **Scalable and less biased language-geometry analysis.** Reduce the cost of constructing pruning-based language vectors, extend experiments to larger and more diverse models, and better isolate model-induced bias in cross-language geometry.

7. **Improved transfer criteria and stochastic robustness certification.** Go beyond distance-only transfer heuristics and tighten control/RL guarantees to distribution-aware, long-horizon stochastic settings.

In summary, the dissertation advances both theory and practice of model compression by establishing a perturbation-centered bridge between parameter-level modifications and task-level behavior. The obtained bounds, algorithms, and empirical findings form a coherent framework for developing compression methods that are not only efficient, but also controllable and reliable.

Bibliography

1. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention Is All You Need. In: Advances in Neural Information Processing Systems. vol. 30. Curran Associates, Inc.; 2017. p. 5998-6008.

2. Brown T, Mann B, Ryder N, Subbiah M, Kaplan JD, Dhariwal P, et al. Language Models Are Few-Shot Learners. In: Advances in Neural Information Processing Systems. vol. 33. Curran Associates, Inc.; 2020. p. 1877-901.

3. LeCun Y, Bengio Y, Hinton G. Deep Learning. Nature. 2015 May;521(7553):436-44.

4. Han S, Mao H, Dally WJ. Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding. arXiv preprint arXiv:151000149. 2015 Oct.

5. Cheng Y, Wang D, Zhou P, Zhang T. A Survey of Model Compression and Acceleration for Deep Neural Networks. arXiv preprint arXiv:171009282. 2017 Oct.

6. Halko N, Martinsson PG, Tropp JA. Finding Structure with Randomness: Probabilistic Algorithms for Constructing Approximate Matrix Decompositions. SIAM review. 2011;53(2):217-88.

7. Frantar E, Alistarh D. SparseGPT: Massive Language Models Can Be Accurately Pruned in One-Shot. In: Proceedings of the 40th International Conference on Machine Learning. PMLR; 2023. p. 10323-37.

8. Cybenko G. Approximation by Superpositions of a Sigmoidal Function. Mathematics of Control, Signals and Systems. 1989 Dec;2(4):303-14.

9. Hornik K. Approximation Capabilities of Multilayer Feedforward Networks. Neural Networks. 1991 Jan;4(2):251-7.

10. Vapnik VN. The Nature of Statistical Learning Theory. New York, NY: Springer; 2000.

11. Goodfellow I, Bengio Y, Courville A. Deep Learning. MIT Press; 2016.

12. Zhang C, Bengio S, Hardt M, Recht B, Vinyals O. Understanding Deep Learning Requires Rethinking Generalization. arXiv preprint arXiv:161103530. 2016 Nov.

13. Neyshabur B, Li Z, Bhojanapalli S, LeCun Y, Srebro N. The Role of Over-Parametrization in Generalization of Neural Networks. International Conference on Learning Representations. 2018 Sep.

14. Nair V, Hinton GE. Rectified Linear Units Improve Restricted Boltzmann Machines. In: Proceedings of the 27th International Conference on Machine Learning (ICML-10); 2010. p. 807-14.

15. He K, Zhang X, Ren S, Sun J. Delving Deep into Rectifiers: Surpassing Human-Level Performance on Imagenet Classification. In: Proceedings of the IEEE International Conference on Computer Vision; 2015. p. 1026-34.

16. Maas AL, Hannun AY, Ng AY, et al. Rectifier Nonlinearities Improve Neural Network Acoustic Models. In: Proc. Icml. vol. 30. Atlanta, GA; 2013. p. 3.

17. Clevert DA, Unterthiner T, Hochreiter S. Fast and Accurate Deep Network Learning by Exponential Linear Units (Elus). arXiv preprint arXiv:151107289. 2015.

18. Hendrycks D, Gimpel K. Gaussian Error Linear Units (Gelus). arXiv preprint arXiv:160608415. 2016.

19. Lecun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. Proceedings of the IEEE. 1998;86(11):2278-324.

20. Hochreiter S, Schmidhuber J. Long Short-Term Memory. Neural Computation. 1997 Nov;9(8):1735-80.

21. Devlin J, Chang MW, Lee K, Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In: Burstein J, Doran C, Solorio T, editors. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). Minneapolis, Minnesota: Association for Computational Linguistics; 2019. p. 4171-86.

22. Bengio Y, Courville A, Vincent P. Representation Learning: A Review and New Perspectives. IEEE Trans Pattern Anal Mach Intell. 2013 Aug;35(8):1798-828.

23. Radford A, Wu J, Child R, Luan D, Amodei D, Sutskever I. Language Models Are Unsupervised Multitask Learners. OpenAI Technical Report. 2019.

24. Mnih V, Kavukcuoglu K, Silver D, Rusu AA, Veness J, Bellemare MG, et al. Human-Level Control through Deep Reinforcement Learning. Nature. 2015;518(7540):529-33.

25. Black K, Brown N, Driess D, Esmail A, Equi M, Finn C, et al. π_0: A Vision-Language-Action Flow Model for General Robot Control. arXiv e-prints. 2024 Oct:arXiv:2410.24164.

26. Bellman R. Dynamic Programming. Science. 1966;153(3731):34-7.

27. Puterman ML. Markov Decision Processes: Discrete Stochastic Dynamic Programming. John Wiley & Sons; 1994.

28. Bertsekas DP. Dynamic Programming and Optimal Control. Athena Scientific; 2005.

29. Sutton RS, Barto AG. Reinforcement Learning: An Introduction. MIT Press; 2018.

30. Gu S, Yang L, Du Y, Chen G, Walter F, Wang J, et al. A Review of Safe Reinforcement Learning: Methods, Theories, and Applications. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2024;46(12):11216-35.

31. Virmaux A, Scaman K. Lipschitz Regularity of Deep Neural Networks: Analysis and Efficient Estimation. Advances in Neural Information Processing Systems. 2018;31.

32. Zhang B, Jiang D, He D, Wang L. Rethinking Lipschitz Neural Networks and Certified Robustness: A Boolean Function Perspective. arXiv preprint arXiv:221001787. 2022 Oct.

33. Nadizar G, Medvet E, Pellegrino FA, Zullich M, Nichele S. On the Effects of Pruning on Evolved Neural Controllers for Soft Robots. In: Proceedings of the Genetic and Evolutionary Computation Conference Companion; 2021. p. 1744-52.

34. Bengio Y, Ducharme R, Vincent P, Janvin C. A Neural Probabilistic Language Model. J Mach Learn Res. 2003 Mar;3:1137-55.

35. Kaplan J, McCandlish S, Henighan T, Brown TB, Chess B, Child R, et al. Scaling Laws for Neural Language Models. arXiv preprint arXiv:200108361. 2020 Jan.

36. Touvron H, Lavril T, Izacard G, Martinet X, Lachaux MA, Lacroix T, et al. Llama: Open and Efficient Foundation Language Models. arXiv preprint arXiv:230213971. 2023.

37. Touvron H, Martin L, Stone K, Albert P, Almahairi A, Babaei Y, et al. Llama 2: Open Foundation and Fine-Tuned Chat Models. arXiv preprint arXiv:230709288. 2023.

38. Jiang AQ, Sablayrolles A, Mensch A, Bamford C, Chaplot DS, de las Casas D, et al. Mistral 7B. arXiv preprint arXiv:231006825. 2023.

39. Grattafiori A, Dubey A, Jauhri A, Pandey A, Kadian A, Al-Dahle A, et al. The Llama 3 Herd of Models. arXiv preprint arXiv:240721783. 2024.

40. Team G, Kamath A, Ferret J, Pathak S, Vieillard N, Merhej R, et al. Gemma 3 Technical Report. arXiv preprint arXiv:250319786. 2025.

41. Ba JL, Kiros JR, Hinton GE. Layer Normalization. arXiv preprint arXiv:160706450. 2016 Jul.

42. Dettmers T, Lewis M, Belkada Y, Zettlemoyer L. Llm. Int8 (): 8-Bit Matrix Multiplication for Transformers at Scale. arXiv preprint arXiv:220807339. 2022.

43. Pope R, Douglas S, Chowdhery A, Devlin J, Bradbury J, Levskaya A, et al. Efficiently Scaling Transformer Inference. arXiv preprint arXiv:160706450. 2022 Nov.

44. Joshi P, Santy S, Budhiraja A, Bali K, Choudhury M. The State and Fate of Linguistic Diversity and Inclusion in the NLP World. In: Jurafsky D, Chai J, Schluter N, Tetreault J, editors. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. Online: Association for Computational Linguistics; 2020. p. 6282-93.

45. Dettmers T, Pagnoni A, Holtzman A, Zettlemoyer L. Qlora: Efficient Finetuning of Quantized Llms. Advances in Neural Information Processing Systems. 2024;36.

46. Frantar E, Ashkboos S, Hoefler T, Alistarh D. Gptq: Accurate Post-Training Quantization for Generative Pre-Trained Transformers. arXiv preprint arXiv:221017323. 2022.

47. Sun M, Liu Z, Bair A, Kolter JZ. A Simple and Effective Pruning Approach for Large Language Models. arXiv preprint arXiv:230611695. 2023.

48. Schmidt E. Zur Theorie Der Linearen Und Nichtlinearen Integralgleichungen: I. Teil: Entwicklung Willkürlicher Funktionen Nach Systemen Vorgeschriebener. Mathematische Annalen. 1907;63(4):433-76.

49. Eckart C, Young G. The Approximation of One Matrix by Another of Lower Rank. Psychometrika. 1936;1(3):211-8.

50. Mirsky L. Symmetric Gauge Functions and Unitarily Invariant Norms. The quarterly journal of mathematics. 1960;11(1):50-9.

51. Horn RA, Johnson CR. Matrix Analysis. Cambridge: Cambridge University Press; 1985.

52. Stewart GW, Sun JG. Matrix Perturbation Theory. San Diego, CA: Academic Press; 1990.

53. Weyl H. Das Asymptotische Verteilungsgesetz Der Eigenwerte Linearer Partieller Differentialgleichungen (Mit Einer Anwendung Auf Die Theorie Der Hohlraumstrahlung). Mathematische Annalen. 1912;71(4):441-79.

54. Jolliffe I. Principal Component Analysis. Springer; 2002.

55. Sirovich L. Turbulence and the Dynamics of Coherent Structures Part Iii: Dynamics and Scaling. Quarterly of Applied Mathematics. 1987;45(3):583-90.

56. Wang Y, Wang H, Zhang SQ. QSVD: Efficient Low-Rank Approximation for Unified Query-Key-Value Weight Compression in Low-Precision Vision-Language Models. arXiv preprint arXiv:251016292. 2025.

57. Lu YC, Yu SF, Weng HH, Wang PS, Hu YF, Hung-Chun L, et al. SkipCat: Rank-maximized Low-Rank Compression of Large Language Models via Shared Projection and Block Skipping. arXiv preprint arXiv:251213494. 2025.

58. Wang Y, Qiao H, Li L, Zhu Q, Che W. CommonKV: Compressing KV Cache with Cross-Layer Parameter Sharing. arXiv preprint arXiv:250816134. 2025.

59. Li L, Qiyuan Z, Wang J, Li W, Gu H, Han S, et al. Sub-MoE: Efficient Mixture-of-Expert Llms Compression via Subspace Expert Merging. arXiv preprint arXiv:250623266. 2025.

60. Chaichana Y, Trachu T, Limkonchotiwat P, Preechakul K, Khandhawit T, Chuangsuwanich E. Decom-Renorm-Merge: Model Merging on the Right Space Improves Multitasking. arXiv preprint arXiv:250523117. 2025.

61. Zhang J, Wiesel A, Haardt M. Low Rank Approximation Based Hybrid Precoding Schemes for Multi-Carrier Single-User Massive MIMO Systems. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP); 2016. p. 3281-5.

62. Nietz AK, Streng ML, Popa LS, Carter RE, Flaherty EB, Aronson JD, et al. To Be and Not to Be: Wide-Field Ca2+ Imaging Reveals Neocortical Functional Segmentation Combines Stability and Flexibility. Cerebral Cortex. 2023 Feb;33(11):6543-58.

63. Zhou J, Wu Y, Liu H, Tian W, Castanon RG, Bartlett A, et al. Human Body Single-Cell Atlas of 3D Genome Organization and DNA Methylation. bioRxiv : the preprint server for biology. 2025.

64. Hall PM, Marshall AD, Martin RR. Incremental Eigenanalysis for Classification. In: BMVC. vol. 98; 1998. p. 286-95.

65. Levy A, Lindenbaum M. Sequential Karhunen-Loeve Basis Extraction and Its Application to Images. In: Proceedings 1998 International Conference on Image Processing. ICIP98 (Cat. No.98cb36269). vol. 2; 1998. p. 456-60.

66. Brand M. Incremental Singular Value Decomposition of Uncertain Data with Missing Values. In: European Conference on Computer Vision. Springer; 2002. p. 707-20.

67. Brand M. Fast Low-Rank Modifications of the Thin Singular Value Decomposition. Linear algebra and its applications. 2006;415(1):20-30.

68. Warmuth MK, Kuzmin D. Randomized Online PCA Algorithms with Regret Bounds That Are Logarithmic in the Dimension. Journal of Machine Learning Research. 2008;9(10):2287-320.

69. Mitliagkas I, Caramanis C, Jain P. Memory Limited, Streaming PCA. In: Proceedings of the 27th International Conference on Neural Information Processing Systems. vol. 2. Curran Associates Inc.; 2013. p. 2886-94.

70. Woodruff DP. Sketching as a Tool for Numerical Linear Algebra. Found Trends Theor Comput Sci. 2014 Oct;10(1–2):1-157.

71. LeCun Y, Denker J, Solla S. Optimal Brain Damage. In: Touretzky D, editor. Advances in Neural Information Processing Systems. vol. 2. Morgan-Kaufmann; 1989. p. 598-605.

72. Hassibi B, Stork D. Second Order Derivatives for Network Pruning: Optimal Brain Surgeon. Advances in neural information processing systems. 1992;5.

73. Molchanov P, Mallya A, Tyree S, Frosio I, Kautz J. Importance Estimation for Neural Network Pruning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2019. p. 11264-72.

74. Yang N, Jang Y, Lee H, Jung S, Jung K. Attribution-Based Task-Specific Pruning for Multi-Task Language Models. arXiv preprint arXiv:220504157. 2022.

75. Ma X, Fang G, Wang X. Llm-Pruner: On the Structural Pruning of Large Language Models. Advances in neural information processing systems. 2024;36.

76. Mishra A, Latorre JA, Pool J, Stosic D, Stosic D, Venkatesh G, et al. Accelerating Sparse Deep Neural Networks. arXiv preprint arXiv:210408378. 2021.

77. Tse D, Viswanath P. Fundamentals of Wireless Communication. Cambridge: Cambridge University Press; 2005.

78. Gramfort A, Luessi M, Larson E, Engemann DA, Strohmeier D, Brodbeck C, et al. MEG and EEG Data Analysis with MNE-python. Frontiers in Neuroscience. 2013;7(267):1-13.

79. McMahan HB, Moore E, Ramage D, Hampson S, y Arcas BA. Communication-efficient Learning of Deep Networks from Decentralized Data. In: Proc.20th International Conference on Artificial Intelligence and Statistics (AISTATS). vol. 54 of Proceedings of Machine Learning Research; 2017. p. 1273-82.

80. Kairouz P, McMahan HB, Avent B, Bellet A, Bennis M, Bhagoji AN, et al. Advances and Open Problems in Federated Learning. Foundations and Trends in Machine Learning. 2021;14(1–2):1-210.

81. Davis C. The Rotation of Eigenvectors by a Perturbation. Journal of Mathematical Analysis and Applications (US). 1963;6.

82. Stewart GW. Perturbation Theory for the Singular Value Decomposition. Digital Repository at the University of Maryland. 1998.

83. Ghaoui LE, Viallon V, Rabbani T. Safe Feature Elimination for the Lasso and Sparse Supervised Learning Problems. arXiv preprint arXiv:10094219. 2010.

84. Fercoq O, Gramfort A, Salmon J. Mind the Duality Gap: Safer Rules for the Lasso. In: Proceedings of the 32nd International Conference on Machine Learning. vol. 37. PMLR; 2015. p. 333-42.

85. Qualcomm AI Research. Massive MIMO Spatial Channel Model Dataset (Dense Urban Scenario), Version 1.0. Qualcomm AI Research Dataset. 2025.

86. Clasen KN, Hackel L, Burgert T, Sumbul G, Demir B, Markl V. reBen: Refined BigEarthNet Dataset for Remote Sensing Image Analysis. In: Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS). IEEE; 2025. p. 1264-8.

87. Takamoto M, Praditia T, Leiteritz R, MacKinlay D, Alesiani F, Pflüger D, et al. PDEBENCH: An Extensive Benchmark for Scientific Machine Learning. Proceedings of the 36th International Conference on Neural Information Processing Systems. 2022.

88. Marafioti A, Zohar O, Farré M, Noyan M, Bakouch E, Cuenca P, et al. SmolVLM: Redefining Small and Efficient Multimodal Models. arXiv preprint arXiv:250405299. 2025.

89. Chaplynskyi D. Introducing UberText 2.0: A Corpus of Modern Ukrainian at Scale. In: Proceedings of the Second Ukrainian Natu-

ral Language Processing Workshop. Dubrovnik, Croatia: Association for Computational Linguistics; 2023. p. 1-10.

90. Raffel C, Shazeer N, Roberts A, Lee K, Narang S, Matena M, et al. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. arXiv e-prints. 2019.

91. Jiao X, Yin Y, Shang L, Jiang X, Chen X, Li L, et al. Tinybert: Distilling Bert for Natural Language Understanding. arXiv preprint arXiv:190910351. 2019.

92. Shamrai M. Language-Specific Pruning for Efficient Reduction of Large Language Models. In: Proceedings of the Third Ukrainian Natural Language Processing Workshop (UNLP)@ LREC-COLING 2024; 2024. p. 135-40.

93. Swadesh M. Lexico-Statistic Dating of Prehistoric Ethnic Contacts: With Special Reference to North American Indians and Eskimos. Proceedings of the American philosophical society. 1952;96(4):452-63.

94. Holman EW, Wichmann S, Brown CH. Automated Dating of Languages. In: Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC 2010). European Language Resources Association; 2011. p. 2052-8.

95. Dryer MS, Haspelmath M, editors. World Atlas of Language Structures. Oxford: Oxford University Press; 2005.

96. O'Horan K, Galle S, Schneider N. Syntactic Variation in Multilingual Representations: Investigating Cross-Lingual Transfer. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Association for Computational Linguistics; 2016. p. 1-11.

97. De Gregorio J, Toral R, Sánchez D. Exploring Language Relations through Syntactic Distances and Geographic Proximity. EPJ Data Science. 2024;13(1):61.

98. Moran S, McCloy D, Wright S. Phonological Similarity and Its Applications in Cross-Linguistic Phonetics. Journal of Phonetics. 2014;42:20-35.

99. Devlin J, Chang MW, Lee K, Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). Association for Computational Linguistics; 2018. p. 4171-86.

100. Conneau A, Khandelwal K, Goyal N, Chaudhary V, Wenzek G, Guzmán F, et al. Unsupervised Cross-Lingual Representation Learning at Scale. In: Jurafsky D, Chai J, Schluter N, Tetreault J, editors. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. Online: Association for Computational Linguistics; 2020. p. 8440-51.

101. Artetxe M, Schwenk H. Massively Multilingual Sentence Embeddings for Zero-Shot Cross-Lingual Transfer and Beyond. Transactions of the Association for Computational Linguistics. 2019;7:597-610.

102. Heffernan K, Çelebi O, Schwenk H. Bitext Mining Using Distilled Sentence Representations for Low-Resource Languages. In: Goldberg Y, Kozareva Z, Zhang Y, editors. Findings of the Association for Computational Linguistics: EMNLP 2022. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics; 2022. p. 2101-12.

103. Rama T, Beinborn L, Eger S. Probing Multilingual BERT for Genetic and Typological Signals. In: Scott D, Bel N, Zong C, editors. Proceedings of the 28th International Conference on Computational Linguistics.

Barcelona, Spain (Online): International Committee on Computational Linguistics; 2020. p. 1214-28.

104. Borg I, Groenen PJ. Modern Multidimensional Scaling: Theory and Applications. Springer Science & Business Media; 2007.

105. Hammarström H, Forkel R, Haspelmath M, Bank S. Glottolog Database 5.1. Zenodo. 2024.

106. Campello RJ, Moulavi D, Sander J. Density-Based Clustering Based on Hierarchical Density Estimates. In: Pei J, Tseng VS, Cao L, Motoda H, Xu G, editors. Advances in Knowledge Discovery and Data Mining. Berlin, Heidelberg: Springer Berlin Heidelberg; 2013. p. 160-72.

107. Lloyd S. Least Squares Quantization in PCM. IEEE Transactions on Information Theory. 1982;28(2):129-37.

108. van der Maaten L, Hinton G. Visualizing Data Using T-SNE. Journal of Machine Learning Research. 2008;9(86):2579-605.

109. McInnes L, Healy J, Melville J. Umap: Uniform Manifold Approximation and Projection for Dimension Reduction. arXiv preprint arXiv:180203426. 2018.

110. Pettie S, Ramachandran V. An Optimal Minimum Spanning Tree Algorithm. Journal of the ACM (JACM). 2002;49(1):16-34.

111. Wikimedia Foundation. Wikimedia Downloads [Website]; 2026. Available from: https://dumps.wikimedia.org.

112. Nguyen T, Nguyen CV, Lai VD, Man H, Ngo NT, Dernoncourt F, et al. CulturaX: A Cleaned, Enormous, and Multilingual Dataset for Large Language Models in 167 Languages. In: Calzolari N, Kan MY, Hoste V, Lenci A, Sakti S, Xue N, editors. Proceedings of the 2024 Joint

International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024). Torino, Italia: ELRA and ICCL; 2024. p. 4226-37.

113. Penedo G, Kydlíček H, Sabolčec V, Messmer B, Foroutan N, Jaggi M, et al. FineWeb2: A Sparkling Update with 1000s of Languages. Hugging Face. 2024 Dec.

114. Rousseeuw PJ. Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis. Journal of computational and applied mathematics. 1987;20:53-65.

115. Hubert L, Arabie P. Comparing Partitions. Journal of classification. 1985;2:193-218.

116. Schütze H, Manning CD, Raghavan P. Introduction to Information Retrieval. vol. 39. Cambridge University Press Cambridge; 2008.

117. Kamada T, Kawai S. An Algorithm for Drawing General Undirected Graphs. Information Processing Letters. 1989;31(1):7-15.

118. Evans L. Measure Theory and Fine Properties of Functions. Routledge; 2018.

119. Barbara NH, Wang R, Manchester IR. On Robust Reinforcement Learning with Lipschitz-Bounded Policy Networks. arXiv preprint arXiv:240511432. 2025 Feb.

120. Kakade S, Langford J. Approximately Optimal Approximate Reinforcement Learning. In: Proceedings of the 19th International Conference on Machine Learning. Morgan Kaufmann Publishers Inc.; 2002. p. 267-74.

Appendix A

List of publications and approbation of results

This appendix contains the list of publications of the PhD candidate on the thesis' topic as well as information about the approbation of the thesis' results.

Scientific works in which the scientific results of the thesis were published:

1. Shamrai M., Analysis of Perturbations of Singular Values in Concatenated Matrices, Ukrainian Mathematical Journal, 77, pp. 1136–1149, 2025, https://doi.org/10.1007/s11253-025-02512-1.
(Scopus – Q3, WoS – Q3, SJR – Q3).

2. Shamrai M., Closed-Form Robustness Bounds for Second-Order Pruning of Neural Controller Policies, Proceedings of the Institute of Applied Mathematics and Mechanics NAS of Ukraine, 39, pp. 81-89, 2025, https://doi.org/10.37069/1683-4720-2025-39-7. (Category B journal).

3. Shamrai M., Nonasymptotic Bounds on Return Degradation for OBD-Pruned Neural Controllers. Bulletin of the Taras Shevchenko National University of Kyiv, Physics and Mathematics, 81(2), pp. 155-158, 2025, https://doi.org/10.17721/1812-5409.2025/2.24
(Scopus – Q4, SJR – Q4).

4. Shamrai M., Concatenated Matrix SVD: Compression Bounds, Incremental Approximation, and Error-Constrained Clustering, 2026, 2601.11626

5. Shamrai M., Hamolia V., Deep Language Geometry: Constructing a Metric Space from LLM Weights, In Proceedings of the 15th International Conference on Recent Advances in Natural Language Processing, pp. 1127–1136, Varna, Bulgaria, 2025, https://aclanthology.org/2025.ranlp-1.130/ (Scopus – Q2)

6. Shamrai M., Language-Specific Pruning for Efficient Reduction of Large Language Models, In Proceedings of the Third Ukrainian Natural Language Processing Workshop (UNLP) @ LREC-COLING 2024, pp. 135–140, Torino, Italia. ELRA and ICCL, 2024, https://aclanthology.org/2024.unlp-1.16.pdf.

Scientific works certifying the approbation of the thesis' materials:

1. Shamrai M., Perturbation Analysis of Singular Values in Concatenated Matrices, Abstracts of the International Conference of Young Mathematicians, Kyiv, The Institute of Mathematics of the National Academy of Sciences of Ukraine, 2025, https://www.imath.kiev.ua/~young/youngconf2025/abstracts/Shamrai.pdf.

2. Shamrai M., Control Error Bound for Pruned Neural Controllers, VIII International Scientific Conference "Modern Problems of Mechanics", Kyiv, Taras Shevchenko National University of Kyiv, 2025, https://drive.google.com/file/d/1OTC7p6qjED_sRUVvGoo8UFQPVUGihQEe.

Information on the approbation of the thesis' results. The main results of the thesis were reported and discussed at:

- Third Ukrainian Natural Language Processing Workshop (UNLP) at LREC-COLING (Torino, Italy, 2024);

- International Conference of Young Mathematicians (Kyiv, The Institute of Mathematics of NAS of Ukraine, 2025);
- VIII International Scientific Conference “Modern Problems of Mechanics” (Kyiv, Taras Shevchenko National University of Kyiv, 2025);
- 15th International Conference on Recent Advances in Natural Language Processing (Varna, Bulgaria, 2025);
- Seminar of Young Scientists (Kyiv, Institute of Mathematics of NAS of Ukraine, 2026).

Appendix B

Full list of languages used in empirical results

Afrikaans, Albanian, Arabic, Aragonese, Armenian, Asturian, Azerbaijani, Bashkir, Basque, Bavarian, Belarusian, Bengali, Bishnupriya Manipuri, Bosnian, Breton, Bulgarian, Burmese, Catalan, Cebuano, Chechen, Chinese (Simplified), Chinese (Traditional), Chuvash, Crimean Tatar, Croatian, Czech, Danish, Dutch, Egyptian Arabic, English, Esperanto, Estonian, Finnish, French, Galician, Georgian, German, Greek, Gujarati, Haitian, Hebrew, Hindi, Hungarian, Icelandic, Ido, Indonesian, Irish, Italian, Japanese, Javanese, Kannada, Kazakh, Kirghiz, Korean, Latin, Latvian, Lithuanian, Lombard, Low Saxon, Luxembourgish, Macedonian, Malagasy, Malay, Malayalam, Marathi, Min Nan Chinese, Minangkabau, Nepali, Newar, Norwegian (Bokmal), Norwegian (Nynorsk), Occitan, Persian (Farsi), Piedmontese, Polish, Portuguese, Punjabi, Romanian, Russian, Scots, Serbian, Serbo-Croatian, Sicilian, Slovak, Slovenian, South Azerbaijani, Spanish, Sundanese, Swahili, Swedish, Tagalog, Tajik, Tamil, Tatar, Telugu, Turkish, Ukrainian, Urdu, Uzbek, Vietnamese, Volapük, Waray-Waray, Welsh, West Frisian, Western Punjabi, Yoruba.

Wikipedia includes all these languages. CulturaX lacks Chinese (Traditional), Min Nan Chinese, Scots, and Crimean Tatar. fineweb-2 does not include Chinese (Traditional), English, Serbo-Croatian, or Tagalog. For the English subset in fineweb-2, we use the fineweb dataset[B.1].

[B.1] https://huggingface.co/datasets/HuggingFaceFW/fineweb

Appendix C

Additional figures of deep language geometry

Figure C.1 displays the MST coloured by k-means clusters. We set $k = 18$ – one cluster for each category plotted in Figure 3.1 (15 natural families plus 3 constructed languages) – so that the cluster colours can be compared directly with the family colours. Most clusters coincide with their expected families, but not all. Notably, Turkish is grouped with Hungarian and Finnish rather than with the other Turkic languages.

Figure C.2 uses HDBSCAN with a minimum cluster size of two. This gives 24 clusters. Crimean Tatar is treated as outlier, while Ukrainian now connects directly to Polish.

Figures C.3 and C.4 give two other views of the same data using t-SNE and UMAP. Like the MST, they highlight clear family groups.

Figure C.5 shows the confusion matrix between k-means clusters and high-level language families. The clusters are first matched to families with the Hungarian algorithm for clearer alignment. Figure C.6 presents the same matrix, but for the finer primary branches of each family.

Figure C.1. MST of all languages. Colours show k-means clusters with $k = 18$ (one cluster for each language family).

Figure C.2. MST of all languages. Colours show HDBSCAN clusters (minimum cluster size = 2). Points marked as outliers by the algorithm are left out.

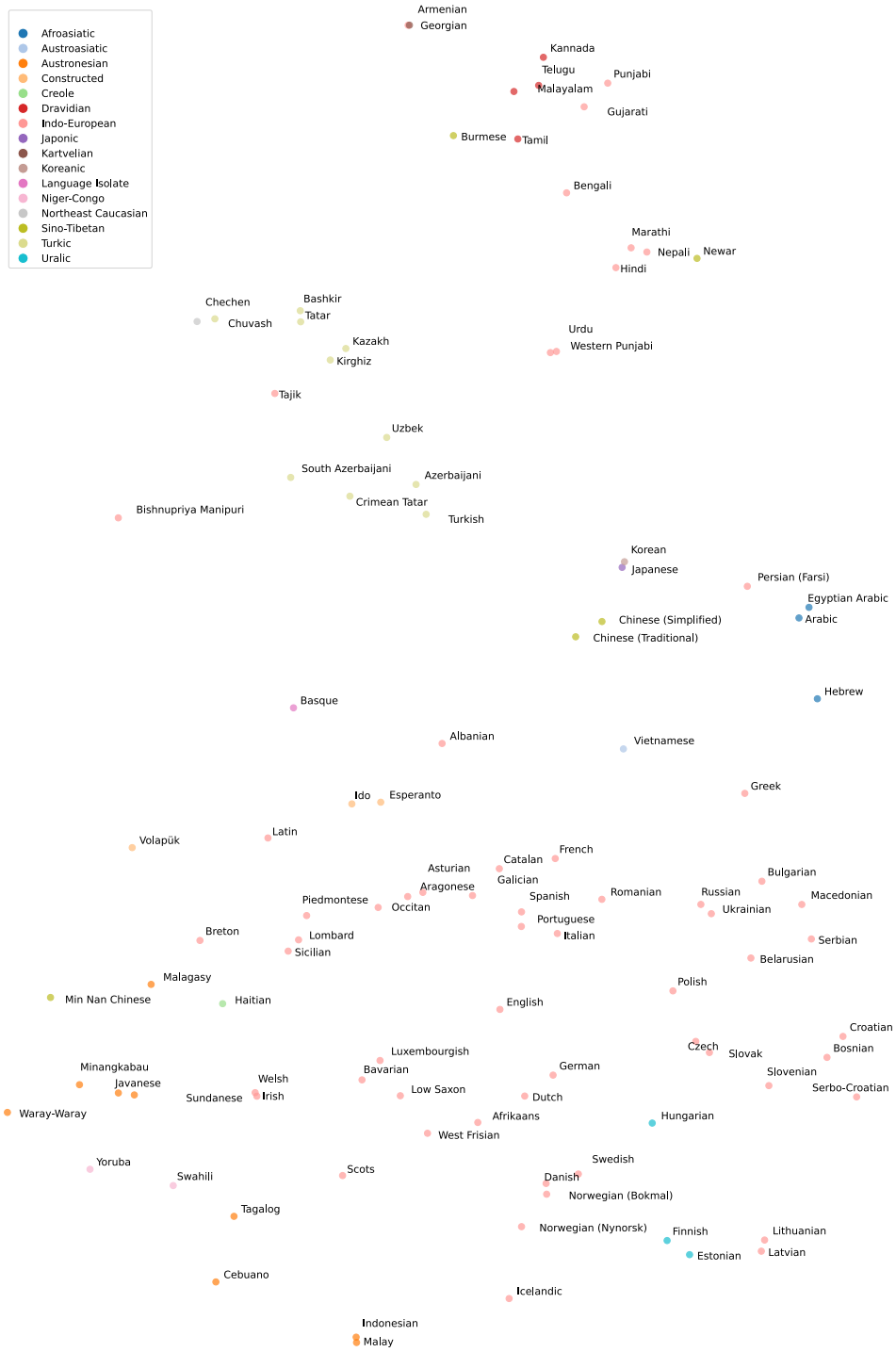


Figure C.3. t-SNE plot of all languages. Colours show language families.

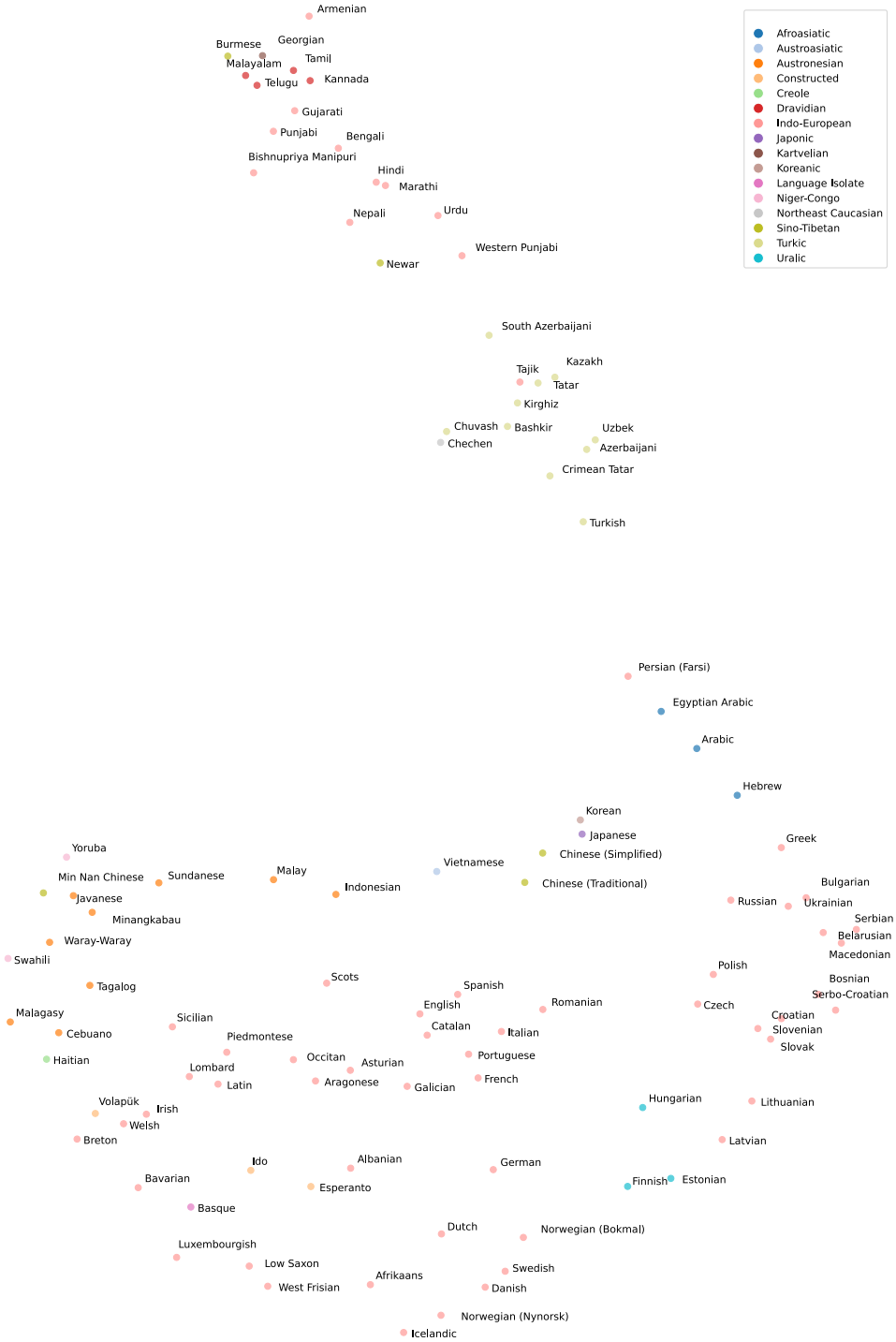


Figure C.4. UMAP plot of all languages. Colours show language families.

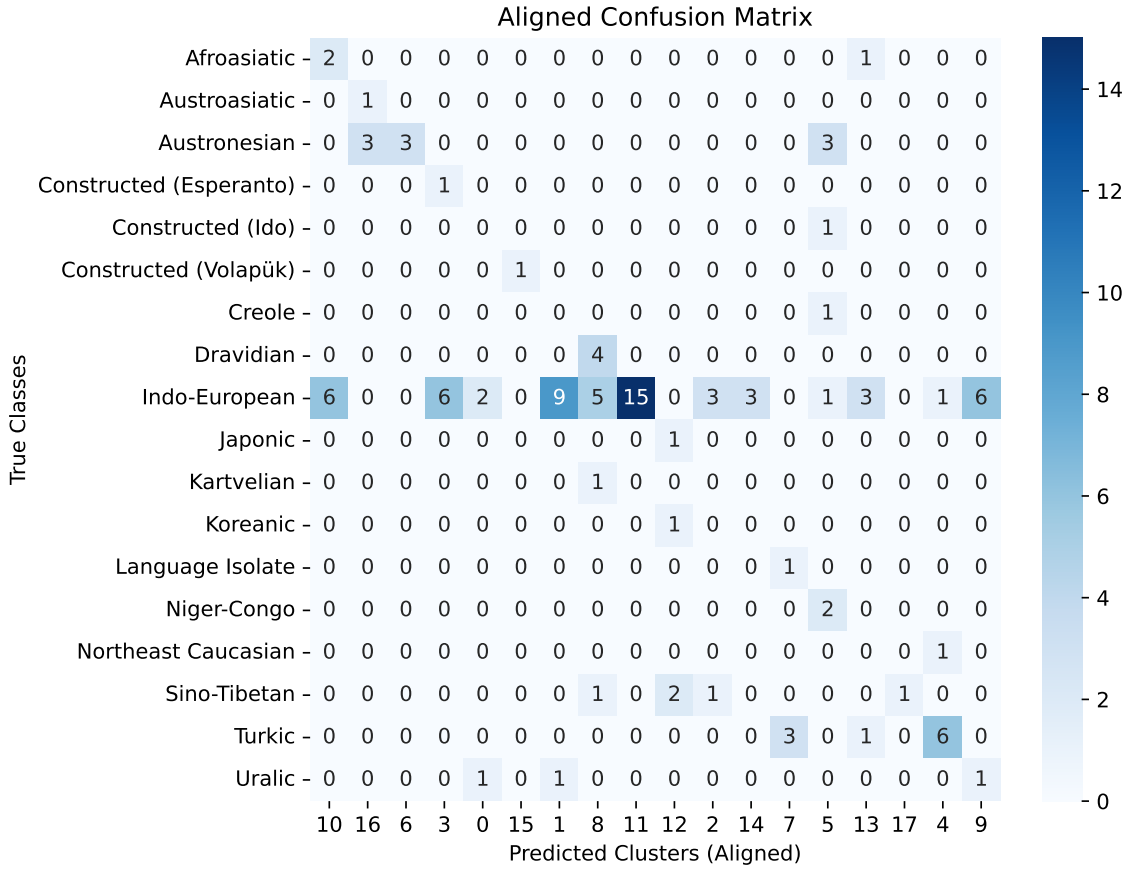


Figure C.5. Adjusted confusion matrix between clusters obtained by k-means and macro families of languages. Number of clusters equal to 18.

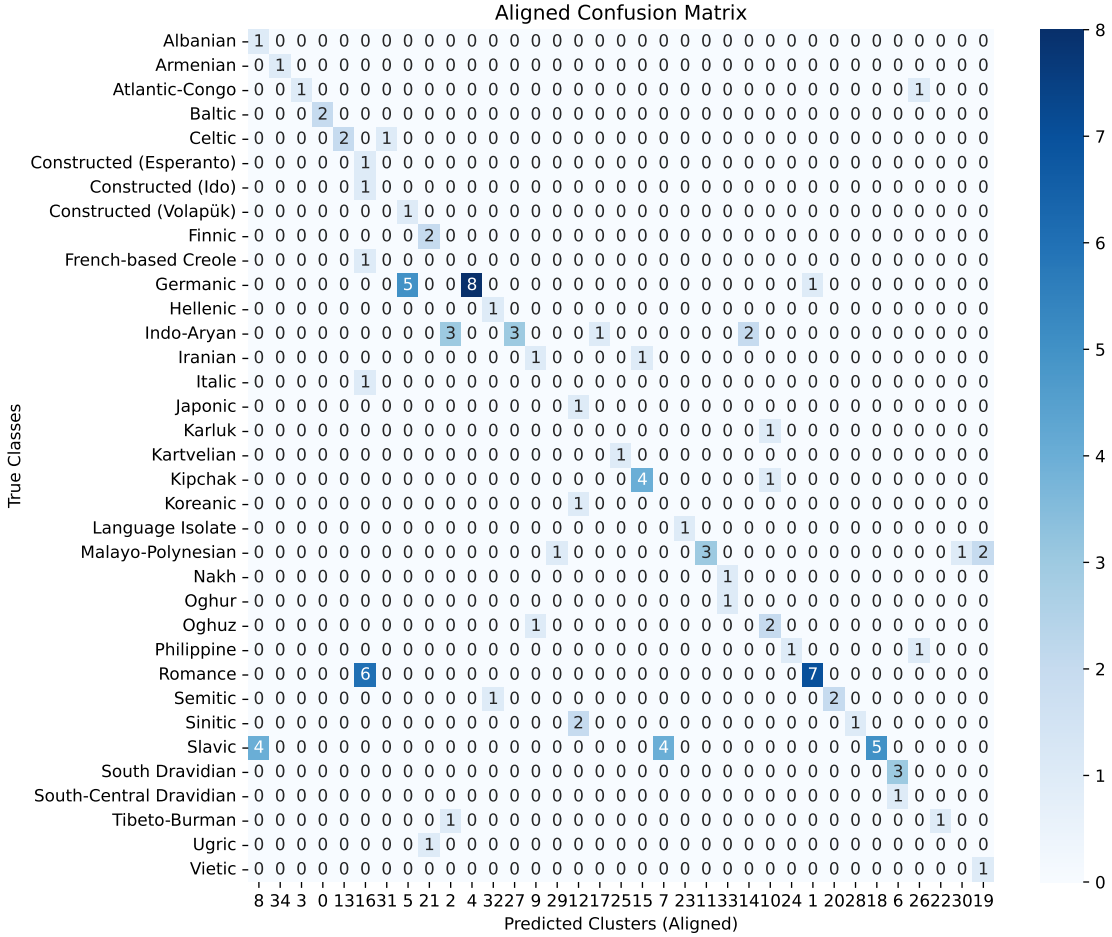


Figure C.6. Adjusted confusion matrix between clusters obtained by k-means and primary branches of language families. Number of clusters equal to 35.