Robust AI-Generated Text Detection by Restricted Embeddings

Anonymous ACL submission

Abstract

Growing amount and quality of AI-generated texts makes detecting such content more difficult. In most real-world scenarios, the domain (style and topic) of generated data and the generator model are not known in advance. In this work, we focus on the robustness of classifier-based detectors of AI-generated text, namely their ability to transfer to unseen generators or semantic domains. We investigate the geometry of the embedding space of Transformer-based text encoders and show that clearing out harmful linear subspaces helps to train a robust classifier, ignoring domain-specific spurious features. We investigate several subspace decomposition and feature selection strategies and achieve significant improvements over state of the art methods in cross-domain and cross-generator transfer. Our best approaches for head-wise and coordinate-based subspace removal increase the mean out-of-distribution (OOD) classification score by 9% and 14% for RoBERTa and BERT embeddings respectively.

1 Introduction

The proliferation of generative AI leads to an explosion in AI-generated content. Large language models (LLMs) can produce text that is very similar to human-written, and image generation models provide increasingly realistic results with more detailed control over generated images. However, AI-generated content can be used for malicious purposes, which leads to the *artificial content detection* (ATD) problem: has a given text or image been created by an AI model or a human? Existing approaches for artificial content detection can be divided into *score-based* and *classifier-based*. The former aim to identify and measure features that distinguish artificial content from real; e.g., generated text may exhibit statistical artifacts due to the specific generation process used by a language model (Gehrmann et al., 2019), the difference may lie in perplexities measured by another language model (Solaiman et al., 2019), curvature of the probability function (Mitchell et al., 2023), or intrinsic dimensionality of contextualized representations (Tulchinskii et al., 2023). However, score-based methods often rely on prior knowledge about a specific generator and/or semantic domain, and known traces may be easy to remove, e.g., by paraphrasing the text (Krishna et al., 2023). One notable exception is the intrinsic dimension feature for text content, shown to be robust to domain transfer and paraphrasing (Tulchinskii et al., 2023), but its overall detection quality is relatively modest.

Supervised classification methods show almost perfect in-domain detection quality, but fail to generalize to unseen text topics and writing styles (Wang et al., 2024b; Tulchinskii et al., 2023). The choice of training data, both artificial and generated, is crucial for successful out-of-domain (OOD) transfer. In general, while usually there exist features that can distinguish between natural and artificial subsets of the training set, the classifier may lock into dataset-specific spurious differences and hence generalize poorly. It is hard to say in advance if a classifier trained on a given dataset will generalize well to new unseen generators and data sources. Previous approaches to OOD detection for ATD include UID-based detectors (Venkatraman et al., 2023) and domain adversarial training (Bhattacharjee et al., 2024), but most of these methods are very data-intensive (Wang et al., 2024a).

In this work, we aim to improve supervised classification by ignoring spurious features to enhance cross-domain robustness, training on small number of domains or generators. Specifically, we focus on methods related to extracting *residual subspaces* and deleting information from embeddings.

Typically, high-dimensional data mainly occupy a few dimensions, and projecting onto a complementary subspace results in lower variance. In many applications, retaining only important directions while treating projections onto less loaded subspaces as residual noise can benefit downstream tasks. However, for tasks such as OOD detection the principal components of a dataset may be the least useful. Kamoi and Kobayashi (2020) found that nullifying the first (least important) principal components in the embedding space fine-tuned on in-domain (ID) data enhances OOD detection quality; this is known as the partial Mahalanobis distance. Podolskiy et al. (2021) conducted similar analysis for Transformer-based text classifiers and found that ID data has orthogonal classes and lies on a unit sphere in a low-dimensional space. The main difference between ID and OOD data lies in the residual subspace, hence the partial Mahalanobis distance performs well in OOD detection.

It is important to note that not all neural networks learn useful residual subspaces for a given dataset; e.g., Podolskiy et al. (2021) and Ren et al. (2021) find that on text, CNN classifiers learn representations where components with low singular values contain too much information about ID data, making it difficult to distinguish OOD examples.

In this work, we apply similar techniques to artificial text detection (ATD), a problem similar to OOD detection. Domain shift, with variations in text styles, topics, and new generation models, is a major challenge for ATD. Supervised classifiers, even performing well on validation datasets, struggle in realistic settings, where the domain and model of the AI-written text are unknown. To address this, we first show that training a classifier on the residual subspace significantly enhances ATD robustness under domain and model shift. Next, we show that controllable subspace removal can improve robustness, while also providing us with interpretable information about AI-written texts and domains. In particular, we use recent advances in concept erasure (Belrose et al., 2023), experimenting with erasing semantic and syntactic concepts based on probing tasks by Conneau et al. (2018); we show that some concepts are harmful for cross-domain and cross-model transfer in ATD.

Our primary contributions are: (i) a first application of the residual subspace approach for robust ATD; we show that restricting the detector to a residual subspace increases cross-topic and cross-model robustness with especially significant improvements on the most difficult samples; (ii) analysis of different *residual decomposition* techniques, such as nullifying head-wise subspaces in intermediate data representations and concept erasure; (iii) analysis of applicability of our methods with different encoder- or decoder-based backbone models. Besides, we create and release an extension for one of the datasets with recent generating model GPT-4-o on three domains. Below, Section 2 surveys related work, Section 3 describes the proposed methods, Section 4 introduces the datasets, Section 5 presents a comprehensive experimental evaluation, and Section 6 concludes the paper.

2 Related Work

Linear subspaces in Transformer-based models are known to represent concepts. Hernandez and Andreas (2021) studied low-dimensional subspaces that encode linguistic features in BERT; linear structure is known for such concepts as truthfulness (Marks and Tegmark, 2023) and sentiment (Tigges et al., 2023). This direction has been extended to the *linear representation hypothesis* that posits that language models operate with one-dimensional representations of concepts in the activation space (Bricken et al., 2023; Park et al., 2023). However, Engels et al. (2024) showed that some concepts are multi-dimensional.

Components of Transformer-based embeddings can provide useful features via the geometry of their inner representations or parameter spaces. For instance, *outlier dimensions* in the embedding spaces of models such as BERT (Devlin et al., 2019) or RoBERTa (Liu et al., 2019), characterized by unusually high variance and/or mean values, have been studied in detail, including their emergence during training and effects of disabling them post-training (Kovaleva et al., 2021), their relationship with positional embeddings and impact on word-in-context tasks (Luo et al., 2021), influence on the quality of representations (Timkey and van Schijndel, 2021), and relations to the shapes of attention maps and token frequencies (Puccetti et al., 2022). Activations of Transformer-based LMs have been investigated for language structure information (Jawahar et al., 2019), semantic and syntactic features (Conneau et al., 2018); the latter work also introduces a comprehensive selection of probing tasks. However, only outlier dimensions have been studied in full detail; we aim to address this gap by studying how removing specific dimensions from RoBERTa embeddings can improve detection of artificially generated text.

Semantics of attention heads in Transformers have been studied for a long time: Kovaleva et al. (2019) provided empirical research on BERT attention heads, demonstrating overparameterization by pruning some of them, Michel et al. (2019) showed that most heads can be removed at test time without significant performance loss. Clark et al. (2019) studied the specialization of attention heads; Pande et al. (2021), their functional roles. In BERT-like models, important information is distributed across layers; e.g., Jawahar et al. (2019) showed that lower layers capture phrase-level information, which is partly lost in the upper layers; the model captures a hierarchy of linguistic information, with deeper layers required to capture long-distance dependencies. Bian et al. (2021) showed that attention maps are correlated across layers and organized into clusters. Therefore, here we focus on groups of attention heads within a layer rather than individual heads.

Artificial text detection (ATD) is a new field of study (up until recently, artificial content was mostly easy to distinguish), but there already exist many promising approaches. Score-based methods include DetectGPT, which measures the curvature of the probability function (Mitchell et al., 2023), and GPTZero (Tian and Cui, 2023), which checks the perplexity and burstiness of a text; these methods, however, are limited to a single domain or generator. Throughout recent work, it remains a reasonable baseline for general and cross-domain ATD to take embeddings from BERT-like models as a feature space and train logistic regression (LR) over them. Following Tulchinskii et al. (2023) and Jawahar et al. (2020), we take the RoBERTa model (Liu et al., 2019) to extract text embeddings, use mean-pooling over embeddings, and train LR models for ATD. The recent SemEval-2024 competition (Wang et al., 2024a) proposed challenge in a multi-generator, multi-domain, and multi-language setting based on the new ATD dataset that was introduced in Wang et al. (2024b). Task 8 included problems such as binary classification, source identification, and fake/real text boundary detection. Solutions used approaches such as LLM fine-tuning (RoBERTa, XLM-R), contrastive learning, and ensemble methods. However, while all these approaches are data-intensive, absolute classification quality is still poor. In this work, we use classifiers that perform well on in-domain data and aim to improve their performance on unseen domains.

3 Methods

Removing unnecessary features is often an effective method to improve the robustness of a machine learning model. The embedding space has linear substructures responsible for linguistic features such as token frequencies, word-in-context information etc. (Luo et al., 2021; Puccetti et al., 2022). We aim to detect and erase such substructures, which are harmful for ATD generalization.

3.1 Linear decompositions of embeddings

PCA and the standard basis. Let $\mathbf{x}$ be some text input, $\mathbf{z} \in \mathbb{R}^d$, its embeddings obtained by some model, $\mathbf{z} = M(\mathbf{x})$, $\mathcal{C} = \{\mathbf{c}_1, \dots, \mathbf{c}_d\}$, a basis of $\mathbb{R}^d$, and let α_i be coefficients of $\mathbf{z}$ in $\mathcal{C}$, $\mathbf{z} = \sum_{i=1}^d \alpha_i \mathbf{c}_i$. We want to split $\mathcal{C}$ into *good* and *bad* parts, $\mathcal{C} = \mathcal{C}_g \cup \mathcal{C}_b$, so that components in $\mathcal{C}_g$ contain most of the information general for all domains, while $\mathcal{C}_b$ is responsible for spurious domain-specific features. Then, we construct a classifier on *restricted* embeddings $\mathbf{z}'$ where the "bad" part is nullified, $\mathbf{z}' = \sum_{i \in \mathcal{C}_g} \alpha_i \mathbf{c}_i$. Intuitively, information about the style, topic, and other semantic properties is harmful for ATD, and we want to focus on residual features that are less important for other NLP tasks. Podolskiy et al. (2021) show that PCA can serve as such a decomposition for a Transformer-based model: removing top components computed for an in-domain dataset improves OOD detection. Indeed, for a dataset of natural texts $\mathcal{D}$, subspace $\langle \mathcal{C}_b \rangle$ should "explain" the data variability, while the variance of $\mathcal{D}$ projected on $\langle \mathcal{C}_g \rangle$ is expected to be low. PCA is a theoretically optimal way to find such subspaces (see Appendix A.1).

Despite PCA's solid theoretical background, in practice it does not always perform well; in ATD, we usually deal with a small dataset that cannot fully capture the real distribution, which is bad for PCA. To access data properties beyond those represented in our train set, we propose to utilize the internal structure of the pretrained embedding model. Indeed, Transformer-based models tend to *disentangle* some data properties during training, and semantic interpretation has been discovered for some neurons and embedding dimensions (Luo et al., 2021; Timkey and van Schijndel, 2021; Puccetti et al., 2022). We hypothesize that such "built-in" disentanglement could lead to meaningful subspaces spanned by a subset of the *standard basis*, i.e., vectors $\{\mathbf{e}_1 = [1, 0, \dots, 0], \mathbf{e}_2 = [0, 1, \dots, 0], \dots, \mathbf{e}_d = [0, 0, \dots, 1]\}$. Projection to a subspace $\langle \mathbf{e}_i | i \in S \rangle$ for some subset of indices S can be done by simply nullifying all embedding dimensions except S. Our experiments support this intuition: PCA-based decomposition does not lead to any significant changes in detector's quality (see Appendix F), while coordinate-based subspace removal significantly improves transfer scores.

Attention heads as linear substructures. Both decompositions discover *global* linear structure, i.e., universal directions in the embedding space independent of input data. But it is much more natural to rely on *local* linearity of the data and try to discover substructures in a data manifold that does not necessarily form global linear subspaces. For text embeddings, the neural network represents a function from $\mathbb{R}^{d\times T}$ to a data manifold $\mathcal{M}$. We can decompose this function into a sum of input-dependent components of the same functional form. Cammarata et al. (2020) proposed *linear circuits*, showing that the data flow in a Transformer can be represented as the main residual stream with linear addition of flows from other elements of the model (attention heads and feed-forward blocks). We are mostly interested in attention flows because it is well known that attention heads in Transformers have highly specialized functions (Kovaleva et al., 2019; Pande et al., 2021), so we hypothesize that head-wise decomposition should reflect the "built-in" disentanglement of the pretrained model. We can represent a Transformer-based embedding as

$$\mathbf{z} = \Pi\left[\alpha(\mathbf{x})\mathbf{x}_0 + \sum_l \beta_l(\mathbf{x})\mathrm{MLP}^l + \sum_l \sum_h \gamma_l(\mathbf{x})A^{l,h}\right], \quad (1)$$

where $A^{l,h}$ are the outputs of attention heads, α, β, γ are scalar functions, and Π is a centering projection $\Pi(\mathbf{x}) = \mathbf{x} - \frac{1}{d}\sum_{i=1}^{d}\mathbf{x}^i$ (see Appendix A.2).

Concept erasure. Finally, we consider an embedding space decomposition based on extracted linear directions or low-rank subspaces responsible for some harmful semantic feature $\mathbf{z}_F$. If such a direction is found, we can remove it by subtracting the component corresponding to this direction from the embedding. Namely, we erase the feature as

$$\hat{\mathbf{z}} = \mathbf{z} - P_F(\mathbf{z}), \quad (2)$$

where P_F is the projection to the subspace $\mathbf{z}_F$.

3.2 Subspace removing methods

Greedy search. Our basic approach chooses the best features using a small subset of domains excluded from the whole dataset. Given a multi-domain dataset $D = D_1 \cup \cdots \cup D_k$, where D_i are domain subsets, we first choose two domains $D_{\text{search}} = \{D_1, D_2\}$ to perform feature selection. On each step, we train a classifier, removing one component, and compare their performance. First, we train ATD classifiers on texts from D_1 only and evaluate on D_2, getting a feature ranking on $D_1 \to D_2$ transfer. Then, we do the opposite, getting a ranking for $D_2 \to D_1$. The final set of residual features is obtained as the intersection of top-score lists in both rankings (see Appendix B.4).

Head pruning removes some components in decomposition (1) by replacing the output of a given head with zeros on inference. Importantly, this approach is approximate because, besides its direct impact as a component in the decomposition, each head also has an indirect influence on all computations on subsequent layers. But Gandelsman et al. (2023) showed that this indirect impact is small and can be ignored (see also Appendix A.2). To choose the set of heads for pruning, we note that different layers contain different kinds of information (e.g., semantic information is mostly in bottom and middle layers), and the linguistic complexity of tasks solved by attention heads grows from bottom to top (Kovaleva et al., 2019; Tenney et al., 2019). Therefore, we simply prune every layer separately.

Concept erasure by probing tasks. To remove a linear subspace responsible for some data properties, we apply a concept erasure technique called LEACE (Belrose et al., 2023). Suppose we have a k-class classification task defined by a dataset Z with one-hot labels Y, and we want to erase all the knowledge required for linear separation of the classes. LEACE is a projection-based method of the form (2), with theoretical guarantees that any linear classifier on top of $\hat{\mathbf{z}}$ cannot solve the classification task better then a constant predictor. Erasing a concept from an embedding $\mathbf{z}$ is defined as

$$\hat{\mathbf{z}} = \mathbf{z} - W^{+}(W\Sigma_{ZY})(W\Sigma_{ZY})^{+}Wz, \quad (3)$$

where $W = (\Sigma_{ZZ}^{1/2})^{+}$, Σ_{ZZ} is Z's covariance matrix, Σ_{ZY} is the cross-covariance of Z and Y. Geometrically, LEACE is the least-squares-optimal transform that maps centroids of different classes of the dataset (Z, Y) to the same point, making linear separation impossible.

In this work, we utilize probing tasks provided by Conneau et al. (2018), designed to represent elementary linguistic concepts (see Section 4). These experiments allow us to not only improve ATD robustness, but also obtain insights about the influence of interpretable linguistic features.

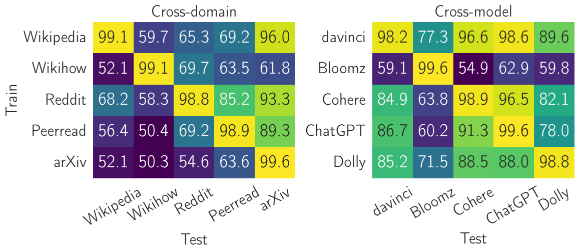

Figure 1: Mean accuracy in cross-domain (left) and cross-model ATD by RoBERTa-base on SemEval

4 Data

ATD datasets. There are few high-quality datasets with both human and artificial text. One such dataset was presented by Wang et al. (2024b) and used in the SemEval-2024 competition[1]; it covers five domains: Wikipedia, Reddit, WikiHow, PeerRead, and arXiv. We have used five text generation models: ChatGPT (Schulman et al., 2022), Davinci003[2], Cohere[3] , Dolly-v2 (Conover et al., 2023), and BLOOMz (Muennighoff et al., 2023). Since the amount of human-written text in each domain is larger than generated by each model, we crop human data so that there are 3000 samples of parallel data for each domain and model/human combination. Similar to Wang et al. (2024b), we create two tasks: (1) in the *cross-domain* task, we concatenate data across generating models, getting five binary ATD tasks in different domains; (2) in the *cross-model* task, concatenation across domains yields five binary ATD tasks for each generator model. Thus, results are presented as 5 × 5 heatmaps (e.g., Fig. 1) and its aggregations.

Our second dataset, used by Tulchinskii et al. (2023), has three domains—Wikipedia, Reddit, StackExchange—with davinci003 generations. Compared to SemEval, it has a larger distribution shift in basic text features (e.g., length), which makes it harder for cross-domain transfer. We extend it by adding similar text generated by GPT-4o: continuing text from Wikipedia articles, long-form question answering on Reddit Q&A and StackExchange. Thus, we obtain a dataset, called below GPT-3D, with six domain-model pairs. For GPT-3D we report average OOD scores, i.e. the accuracy of classifiers trained on one domain-model subset and evaluated on the rest; average accuracy values do not include training sets.

Domains	Wikipedia	WikiHow	Reddit	PeerRead	arXiv
Avg. transfer to:	57.2	54.7	64.7	70.4	85.1
Avg. transfer from:	72.5	61.8	76.3	66.3	55.2
Avg. sent. length	38.7	44.4	17.0	14.7	10.4
Avg. “!” count	0.24	0.79	0.25	0.08	0.01
Avg. “?” count	0.12	0.90	0.36	0.43	0.03

Generators	davinci	Bloomz	Cohere	ChatGPT	Dolly
Avg. transfer to:	79.0	68.2	82.8	86.5	77.4
Avg. transfer from:	90.5	59.2	81.8	79.1	83.3
Avg. sent. length	17.7	10.9	15.3	22.6	21.2
Avg. “!” count	0.18	0.50	0.04	0.23	0.29
Avg. “?” count	0.08	0.39	0.11	0.13	0.22

Table 1: Average RoBERTa detector accuracy by domains and by generators on *SemEval* (in %), avg length of generated sentences (in symbols) and avg counts of “!” and “?” marks per text sample; dark red – smallest value in a row, dark green – largest value, red – domains and generators with lowest transfer accuracy.

Probing datasets. For probing and concept erasure experiments, we use the dataset used by Conneau et al. (2018) with several supervised classification tasks: *SentLen*, predicting the length of the sentence, *TreeDepth*, finding the depth of a syntactic tree, *TopConst*, classifying the high-level syntactic structure (top two nodes in the syntax tree), classifying *Tense*, *SubjNum* (subject number), and *ObjNum* (object number) in the main clause, detecting errors with *BShift* (bigram shift, word order inversion in a bigram), *SOMO* (Semantic Odd Man Out, where a word is replaced with a random grammatically fitting word), and *CoordInv* (Coordination Inversion, whether the coordination of two clauses in the sentence is inverted), and predicting exact words from a 1000-word vocabulary in *WC* (Word Content).

5 Results and Analysis

Experimental setup is detailed in the Appendix B. PCA-based results are reported in Appendix F.

Baseline RoBERTa. As a baseline we use logistic regression (LR) trained on mean-pooled RoBERTa embeddings. Results are shown in Fig. 1 for SemEval and Fig. 3a for GPT-3D; the cross-domain and cross-model settings are challenging in both tasks. Fig. 1 shows that in-domain classification is almost perfect for baseline LR on RoBERTa embeddings, but the cross-domain part is very inconsistent: e.g., transfer from Reddit to PeerRead

[1] https://semeval.github.io/SemEval2024/
[2] https://platform.openai.com/docs/models
[3] https://docs.cohere.com/docs/models

works well across all models (91% avg accuracy) but transfer from arXiv to WikiHow is uniformly bad (54%). In *SemEval*, *Wikihow* is the hardest domain to transfer to, while *Arxiv* is the hardest domain to transfer from (Table 1); both domains contain syntactic anomalies (very few or many "!" and "?" marks, unusual average sentence lengths etc.). *Bloomz* is the hardest model to transfer both to and from (Table 1), and it also generates unusual texts (very short sentences replete with "!" and "?"). But generally, it is not easy to predict which transfer direction is easier in ATD or explain the reasons for it; e.g., *Wikipedia*, often used for NLP model evaluation (Merity et al., 2016), is far from the best basis for transfer, especially in the cross-model setting (Fig. 3a). We also compare (Fig. 3f) our proposed methods with the approach based on the intrinsic dimensionality (PHD) of real and artificial texts tokens embedding point clouds, according to (Tulchinskii et al., 2023).

Average transfer results. Table 3 and Fig. 3 show that our methods provide a stable improvement of OOD scores for classifiers trained on separate domain-model subsets, for both SemEval and GPT-3D datasets. *TopConst* concept erasure yields the highest increase among methods that do not have access to OOD data (+3%), and improvement increases for the most difficult domain pairs (e.g., +6% for *Wikipedia–Reddit*). Interestingly, the PHD method by Tulchinskii et al. (2023), while providing very stable cross-domain results for GPT-3-based generations, completely fails to deal with GPT-4o (Fig. 3 (f)), while our methods increase cross-model scores up to 10%. Still, results for the most difficult pairs are unsatisfactory, falling below the random baseline; the only method that can achieve at least random level for *any* OOD subset is head pruning, where the heads are selected on validation set combined of all models and domains examples (+9.1% "cross-all" compared to full RoBERTa, Fig. 3 (e)).

Head pruning for transfer tasks. We adapt head pruning (Voita et al., 2019) to remove a whole layer of attention heads. Since layers of a model have rough linguistic meanings (Jawahar et al., 2019), thus we analyse the impact of structural-level information on ATD. Fig. 2 and Table 2 show detailed results for each layer pruned on *SemEval*. Removing the first layer improves average cross-domain accuracy by 3%, but the improvement is unstable (from −7.1% to +18.9% in different domains). Pruning layers 3 and 4 is more stable and beneficial in both settings. Cross-domain ATD is more challenging; Fig. 2 (top) shows that some domains (*Wikipedia* and *WikiHow*) exhibit similar patterns but others are unrelated. The best scores are in transfer from *Reddit*, achieving 81% mean balanced accuracy with 0-th layer pruned (+5% to full RoBERTa). The cross-model setting is easier and not greatly affected by pruning layers, with the exception of BLOOMz. Here the best source model is GPT-3.5-davinci, with 92% cross-model accuracy after removing layer 4.

	Cross-domain			**Cross-model**		
	Avg	Max ↑	Max ↓	Avg	Max ↑	Max ↓
RoBERTa	73.0	-	-	82.8	-	-
1	**76.0**	**18.9**	-7.1	82.6	**4.4**	-4.4
2	73.9	6.3	-3.7	83.3	2.8	-2.4
3	75.0	8.6	-1.9	83.1	2.6	-1.8
4	74.6	8.4	-1.6	**83.7**	3.3	-1.6
5	73.7	3.6	-1.8	82.9	1.7	-1.4
6	72.6	2.8	-3.7	82.7	1.6	-2.1
7	72.3	1.3	-5.2	82.5	1.2	-3.0
8	73.3	3.5	-3.5	82.5	0.4	-1.8
9	73.1	4.5	**-1.5**	82.7	0.6	-1.2
10	72.7	3.2	-3.2	82.4	0.4	-2.2
11	73.2	3.8	-6.5	82.3	0.4	-1.4
12	73.7	7.2	-3.7	82.8	1.7	**-1.1**

Table 2: Balanced accuracy for OOD classification for different pruned layers on *SemEval*

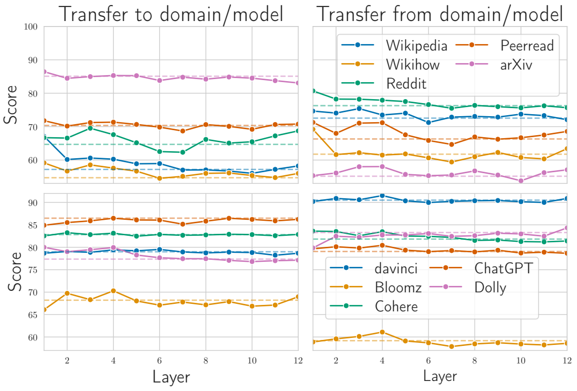

Figure 2: Mean accuracy on *SemEval* with pruned RoBERTa layers. Dashed lines show the baseline.

Concept erasure. Generally, results on *SemEval* show that the best concepts to erase are *TopConst* and *TreeDepth*, improving up to 2.1% on cross-domain transfer and not hurting the cross-model transfer. Erasing *WC* also performs well but is less stable. Figs. 4 give more detailed information. Although changes compared to Table 6 are marginal on average, they range from −8.5% to +13% across domains and models. Grammatical properties, (*Tense*, *SubjNum*, *ObjNum*) have no significant impact, while erasing global syntax information (*TopConst*, *TreeDepth*) improves cross-domain transfer up to $+13\%$, especially from *wikipedia* and *arxiv*. This means that LLMs in general are not good in mimicking complicated syntactic structures, but have no problem with local grammatical categories. Erasing *WC* erasure leads to the largest cross-domain improvement, which means that word semantics produce domain-specific spurious features that harm generalization. There is one outlier: *wikihow*→*arxiv*; we hypothesize that these domains have common word-level features due to many bullet points, numbered lists, and sequential structures in both. For cross-model transfer, erasing all three tasks related to error detection in sentence structure (*BShift*, *CoordInv*, *SOMO*) are harmful for ATD performance and robustness; erasing global syntax (*TopConst*, *TreeDepth*) improves performance, while word content (*WC*) leads to contradictory results.

We conclude that the ability to detect grammatically correct sentences is crucial for robust AI-generated text detection; the difference in global syntax between natural and generated texts is significant, but varies between models and domains, so erasing this information helps generalization, and individual word semantics is a source of spurious features. On the other hand, world-level grammatical categories are captured well by all generators and do not influence ATD performance.

Selecting embedding components and heads. To evaluate component removal, we use *Reddit* and *Wikipedia* domains from GPT-3D as D_{search} (because they have the lowest cross-domain ATD accuracy). For head selection, we used a lay-off evaluation set with samples of all generators and domains from GPT-3D. We evaluate the methods on GPT-3D and *SemEval*, using the same set of removed heads or components. Fig. 3 and Table 3 show the results; transfer to and from *Wikipedia* and *Reddit* subsets has improved. Head Selection greatly improves the performance on validation domains, achieving the best scores among all the methods. Results of cross-task transfer (from GPT-3D to SemEval, Appendix C) indicate that component and head removal works better if components are chosen on the same data distribution where the classifier is trained; still, in general cross-dataset transfer here is not better than the baseline.

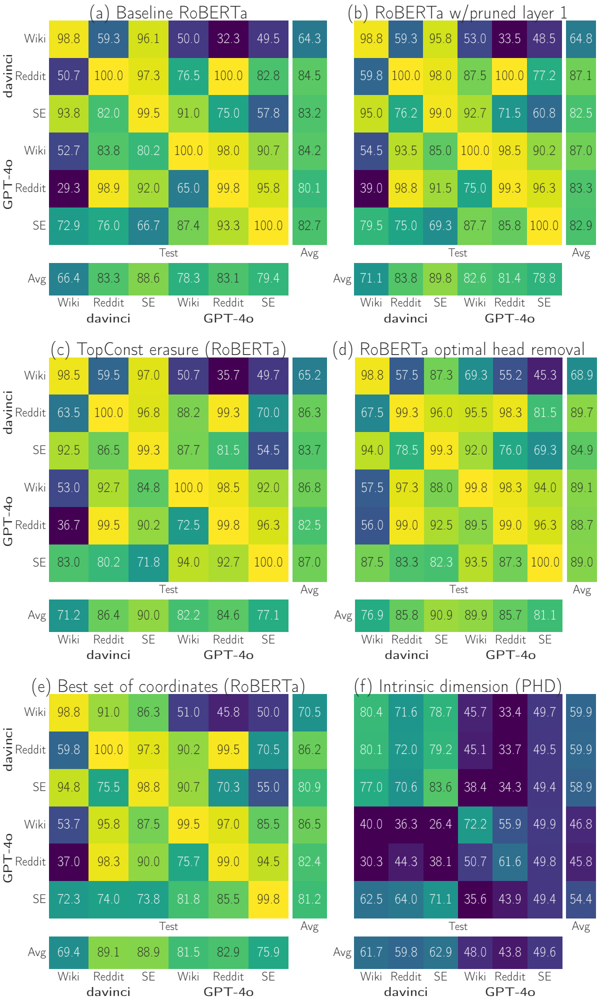

Figure 3: Mean accuracy in cross-domain/cross-model ATD on GPT-3D by: (a) RoBERTa-base, (b) RoBERTa-base with all attention heads pruned from layer 1, (c) RoBERTa with TopConst concept erasure, (d) optimal head removal, (e) best set of coordinates, (f) classifier based on PHD intrinsic dimensions.

	SemEval		**GPT-3D**		
RoBERTa	CD	CM	CD	CM	CA
Baseline	73.0	82.8	84.1	71.0	70.1
Layer 1	**76.0**	82.6	84.8	72.7	72.9
Layer 4	74.6	**83.7**	84.9	72.3	72.0
TopConst erased	75.1	83.1	**86.7**	71.4	73.1
TreeDepth erased	73.9	83.0	85.3	73.3	72.0
Selected heads	74.3	80.0	86.6	**79.3**	**79.2**
Selected coordinates	74.5	82.6	85.4	71.9	72.8

Table 3: Balanced accuracy for OOD classification: cross-domain (CD), cross-model (CM), cross-all (CA).

Influence of the embedding model. RoBERTa is commonly used as the encoder for ATD (Krishna et al., 2023; Solaiman et al., 2019; Tulchinskii et al., 2023), but we have tested other models as well. Table 4 and Figure 14 in the Appendix F show the results; in all cases, we trained LR on mean-pooled embeddings of the last layer. There is an interesting difference between encoder and decoder-based

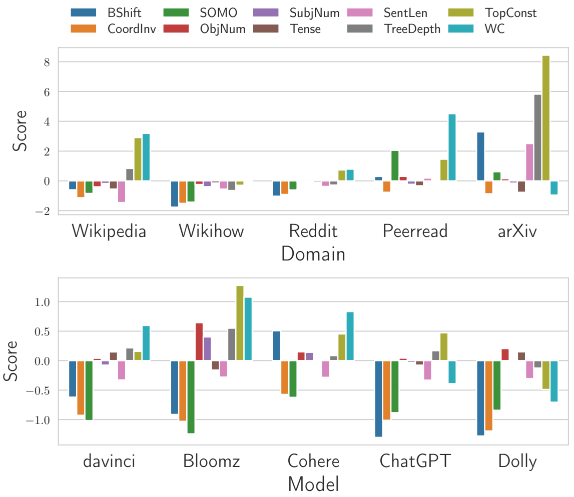

Figure 4: Score change after concept erasure in cross-domain and cross-model settings on *SemEval*.

	BERT, GPT-3D			**Phi2, GPT-3D**		
	CD	CM	CA	CD	CM	CA
Baseline	82.4	81.9	71.1	92.2	**92.3**	86.7
Layer 1	83.2	77.8	69.3	85.5	89.5	78.0
Layer 4	82.2	78.9	69.6	92.6	**92.3**	**87.2**
Selected heads	85.4	81.0	73.1	-	-	-
TopConst erased	83.1	81.4	70.9	91.8	91.5	86.1
TreeDepth erased	84.0	83.2	71.8	**93.3**	91.8	87.0
Selected coordinates	**92.1**	**88.0**	**85.2**	93.1	89.9	86.7

Table 4: Aggregated OOD scores for BERT and Phi2 embeddings. Best results are given in bold.

models: although the quality is very different and correlates with model size, all tested encoders are well suited for our content removal methods (their performance increases, often significantly), while the decoder's behaviour is the opposite. Table 4 shows the results of different subspace removal methods for BERT (Devlin et al., 2019) and Phi-2 (Abdin et al., 2023) embeddings; the latter model is larger, so its baseline scores are much higher, but embedding restriction does not lead to any improvements while BERT's quality increases, making the results of these models comparable after component removal despite the differences in model size. We believe that the different behaviour of our methods reflects the fundamental difference in the embedding space geometry of encoders and decoders caused by limitations of the expressive power of the attention due to the triangular attention mask (e.g., the group of upper triangular matrices does not contain any nontrivial rotations or orthogonal transforms in general). On the other hand, high performance of our methods for relatively small encoder-based models shows that their text representations contain disentangled elementary features learned in pretraining and expressed by separate embedding coordinates, attention heads (i.e., linear terms in input-dependent embedding decompositions), or global directions in the embedding space.

We also report how removing components influences the geometry of the embedding space. PHD intrinsic dimension has the opposite behaviour in GPT-3 and GPT-4 models families: the generalization ability of a PHD-based ATD classifier decreases after removing embedding components (see Appendix E).

6 Conclusion

In this work, we aim to improve the robustness of artificial text detectors via linear feature removal from text embeddings. We propose three ideas that are extremely easy to implement and achieve stable improvement in robustness averaged across domains and models, up to 14% depending on the text encoder. More importantly, we conclude with the following novel insights from our work.

First, new generation models can completely break detectors; e.g., on the GPT-4 family previous detectors' perform below random, while on the same model classifiers demonstrate very high performance in the cross-domain setting. The reason could be the presence of watermarks in GPT-4 generations; if so, watermarks unknown for ATD developers are dangerous, leading to unpredictable black-box behaviour.

Second, performance with respect to the training subset is often counterintuitive; e.g., a classifier trained on Wikipedia may perform worse than on Reddit, although Wikipedia is considered a cleaner domain, better suited for general-purpose models.

Third, Transformer encoders learn disentangled intrinsic features in coordinates and attention heads, and simple decompositions perform better for ATD than more complex approaches. But this effect is less pronounced for decoder models. We plan to study differences in the geometry of encoder and decoder-based text representations in future work.

Finally, global syntax and sentence complexity is a key point for ATD, but the exact differentiating features are domain- and model-specific, so this information should be ignored. Local grammatical categories do not provide an important signal for ATD. Instead, the classifier should rely on features for detecting various types of inconsistencies.

7 Limitations

In this work we demonstrate how state of the art ATD methods can fail, for instance, to transfer to new generative models. Our method increases OOD performance on some generators, but there is no guarantee that this property will be preserved for all future models. Novel pretraining techniques, data collection and processing paradigms, and model architectures can completely change the picture. Since our method is based on supervised classification, it is not clear which features it actually uses for classification. It can also lead to unexpected results, especially in the presence of the so-called watermarks, i.e., small changes in data distribution inside each generated sample deliberately injected by generative model developers. We believe that for truly reliable ATD detection, all conclusions should be interpretable, so that a human analyst could inspect the decision. By the proposed methods of concept erasure, we have made a step towards such kind of ATD approaches.

We have tested our approaches using relatively small subsets of uni-model or uni-domain data and demonstrated promising quality improvements. Nevertheless, it is still not identical to real-world scenarios, where at least several domains and generators are available in training time, and even more have to be considered during the model's application. One of our objectives in this work has been to propose a novel direction that can significantly improve ATD methods in the future and make them more reliable, but currently it is not yet a fully practical production-ready solution.

Finally, we do not address the real-word case of post-processed and paraphrased generations, and also texts partially written by humans. For example, if some sentences of this section have been generated by GPT-4o but then partially corrected by the authors, most probably the methods considered in this work would not be able to detect it. We leave this direction for further study.

References

Marah Abdin, Jyoti Aneja, Sebastien Bubeck, Caio César Teodoro Mendes, Weizhu Chen, Allie Del Giorno, Ronen Eldan, Sivakanth Gopi, Suriya Gunasekar, Mojan Javaheripi, Piero Kauffmann, Yin Tat Lee, Yuanzhi Li, Anh Nguyen, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Michael Santacroce, Harkirat Singh Behl, Adam Taumann Kalai, Xin Wang, Rachel Ward, Philipp Witte, Cyril Zhang, and Yi Zhang. 2023. Phi-2: The surprising power of small language models.

Nora Belrose, David Schneider-Joseph, Shauli Ravfogel, Ryan Cotterell, Edward Raff, and Stella Biderman. 2023. LEACE: Perfect linear concept erasure in closed form. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Amrita Bhattacharjee, Raha Moraffah, Joshua Garland, and Huan Liu. 2024. Eagle: A domain generalization framework for ai-generated text detection. *arXiv preprint arXiv:2403.15690*.

Yuchen Bian, Jiaji Huang, Xingyu Cai, Jiahong Yuan, and Kenneth Church. 2021. On attention redundancy: A comprehensive study. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 930–945, Online. Association for Computational Linguistics.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, et al. 2023. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, page 2.

Nick Cammarata, Shan Carter, Gabriel Goh, Chris Olah, Michael Petrov, Ludwig Schubert, Chelsea Voss, Ben Egan, and Swee Kiat Lim. 2020. Thread: Circuits. *Distill*. Https://distill.pub/2020/circuits.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does BERT look at? an analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy. Association for Computational Linguistics.

Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126–2136, Melbourne, Australia. Association for Computational Linguistics.

Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. 2023. Free dolly: Introducing the world's first truly open instruction-tuned llm.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Joshua Engels, Isaac Liao, Eric J Michaud, Wes Gurnee, and Max Tegmark. 2024. Not all language model features are linear. *arXiv preprint arXiv:2405.14860*.

Yossi Gandelsman, Alexei A Efros, and Jacob Steinhardt. 2023. Interpreting clip's image representation via text-based decomposition. In *The Twelfth International Conference on Learning Representations*.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander M Rush. 2019. Gltr: Statistical detection and visualization of generated text. *arXiv preprint arXiv:1906.04043*.

Evan Hernandez and Jacob Andreas. 2021. The low-dimensional linear geometry of contextualized word representations. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 82–93, Online. Association for Computational Linguistics.

Ganesh Jawahar, Muhammad Abdul-Mageed, and Laks V. S. Lakshmanan. 2020. Automatic detection of machine generated text: A critical survey. In *COLING*, pages 2296–2309.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.

Ryo Kamoi and Kei Kobayashi. 2020. Why is the mahalanobis distance effective for anomaly detection? *arXiv preprint arXiv:2003.00402*.

Olga Kovaleva, Saurabh Kulshreshtha, Anna Rogers, and Anna Rumshisky. 2021. BERT busters: Outlier dimensions that disrupt transformers. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3392–3405, Online. Association for Computational Linguistics.

Olga Kovaleva, Alexey Romanov, Anna Rogers, and Anna Rumshisky. 2019. Revealing the dark secrets of bert. In *Conference on Empirical Methods in Natural Language Processing*.

Kalpesh Krishna, Yixiao Song, Marzena Karpinska, John Wieting, and Mohit Iyyer. 2023. Paraphrasing evades detectors of ai-generated text, but retrieval is an effective defense. *arXiv preprint arXiv:2303.13408*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Ziyang Luo, Artur Kulmizev, and Xiaoxi Mao. 2021. Positional artefacts propagate through masked language model embeddings. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5312–5327, Online. Association for Computational Linguistics.

Samuel Marks and Max Tegmark. 2023. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets. *arXiv preprint arXiv:2310.06824*.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *Preprint*, arXiv:1609.07843.

Paul Michel, Omer Levy, and Graham Neubig. 2019. Are sixteen heads really better than one? In *Neural Information Processing Systems*.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D Manning, and Chelsea Finn. 2023. Detectgpt: Zero-shot machine-generated text detection using probability curvature. *arXiv preprint arXiv:2301.11305*.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng-Xin Yong, Hailey Schoelkopf, Xiangru Tang, Dragomir Radev, Alham Fikri Aji, Khalid Almubarak, Samuel Albanie, Zaid Alyafeai, Albert Webson, Edward Raff, and Colin Raffel. 2023. Crosslingual generalization through multitask finetuning. *Preprint*, arXiv:2211.01786.

Madhura Pande, Aakriti Budhraja, Preksha Nema, Pratyush Kumar, and Mitesh M Khapra. 2021. The heads hypothesis: A unifying statistical approach towards understanding multi-headed attention in bert. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 13613–13621.

Kiho Park, Yo Joong Choe, and Victor Veitch. 2023. The linear representation hypothesis and the geometry of large language models. In *Causal Representation Learning Workshop at NeurIPS 2023*.

Alexander Podolskiy, Dmitry Lipin, Andrey Bout, Ekaterina Artemova, and Irina Piontkovskaya. 2021. Revisiting mahalanobis distance for transformer-based out-of-domain detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 13675–13682.

Giovanni Puccetti, Anna Rogers, Aleksandr Drozd, and Felice Dell'Orletta. 2022. Outlier dimensions that disrupt transformers are driven by frequency. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 1286–1304, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jie Ren, Stanislav Fort, Jeremiah Liu, Abhijit Guha Roy, Shreyas Padhy, and Balaji Lakshminarayanan. 2021. A simple fix to mahalanobis distance for improving near-ood detection. *arXiv preprint arXiv:2106.09022*.

William Rudman, Nate Gillman, Taylor Rayne, and Carsten Eickhoff. 2022. IsoScore: Measuring the uniformity of embedding space utilization. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3325–3339, Dublin, Ireland. Association for Computational Linguistics.

John Schulman, Barret Zoph, Christina Kim, Jacob Hilton, Jacob Menick, Jiayi Weng, Juan Felipe Ceron Uribe, Liam Fedus, Luke Metz, Michael Pokorny, Rapha Gontijo Lopes, Shengjia Zhao, Arun Vijayvergiya, Eric Sigler, Adam Perelman, Chelsea Voss, Mike Heaton, Joel Parish, Dave Cummings, Rajeev Nayak, Valerie Balcom, David Schnurr, Tomer Kaftan, Chris Hallacy, Nicholas Turley, Noah Deutsch, Vik Goel, Jonathan Ward, Aris Konstantinidis, Wojciech Zaremba, Long Ouyang, Leonard Bogdonoff, Joshua Gross, David Medina, Sarah Yoo, Teddy Lee, Ryan Lowe, Dan Mossing, Joost Huizinga, Roger Jiang, Carroll Wainwright, Diogo Almeida, Steph Lin, Marvin Zhang, Kai Xiao, Katarina Slama, Steven Bills, Alex Gray, Jan Leike, Jakub Pachocki, Phil Tillet, Shantanu Jain, Greg Brockman, Nick Ryder, Alex Paino, Qiming Yuan, Clemens Winter, Ben Wang, Mo Bavarian, Igor Babuschkin, Szymon Sidor, Ingmar Kanitscheider, Mikhail Pavlov, Matthias Plappert, Nik Tezak, Heewoo Jun, William Zhuk, Vitchyr Pong, Lukasz Kaiser, Jerry Tworek, Andrew Carr, Lilian Weng, Sandhini Agarwal, Karl Cobbe, Vineet Kosaraju, Alethea Power, Stanislas Polu, Jesse Han, Raul Puri, Shawn Jain, Benjamin Chess, Christian Gibson, Oleg Boiko, Emy Parparita, Amin Tootoonchian, Kyle Kosic, and Christopher Hesse. 2022. Introducing chatgpt.

Haipeng Shen and Jianhua Z Huang. 2008. Sparse principal component analysis via regularized low rank matrix approximation. *Journal of multivariate analysis*, 99(6):1015–1034.

Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, Gretchen Krueger, Jong Wook Kim, Sarah Kreps, et al. 2019. Release strategies and the social impacts of language models. *arXiv preprint arXiv:1908.09203*.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy. Association for Computational Linguistics.

Edward Tian and Alexander Cui. 2023. Gptzero: Towards detection of ai-generated text using zero-shot and supervised methods.

Curt Tigges, Oskar John Hollinsworth, Atticus Geiger, and Neel Nanda. 2023. Linear representations of sentiment in large language models. *CoRR*, abs/2310.15154.

William Timkey and Marten van Schijndel. 2021. All bark and no bite: Rogue dimensions in transformer language models obscure representational quality. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4527–4546, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Eduard Tulchinskii, Kristian Kuznetsov, Kushnareva Laida, Daniil Cherniavskii, Sergey Nikolenko, Evgeny Burnaev, Serguei Barannikov, and Irina Piontkovskaya. 2023. Intrinsic dimension estimation for robust detection of AI-generated texts. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Saranya Venkatraman, Adaku Uchendu, and Dongwon Lee. 2023. Gpt-who: An information density-based machine-generated text detector. *arXiv preprint arXiv:2310.06202*.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. 2019. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5797–5808, Florence, Italy. Association for Computational Linguistics.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, et al. 2024a. Semeval-2024 task 8: Multidomain, multimodel and multilingual machine-generated text detection. *arXiv preprint arXiv:2404.14183*.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Toru Sasaki, Thomas Arnold, Alham Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024b. M4: Multi-generator, multi-domain, and multilingual black-box machine-generated text detection. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1369–1407, St. Julian's, Malta. Association for Computational Linguistics.

A Residual subspaces for ATD

A.1 Formal definitions and theory

In this subsection, we introduce formal definitions and recap some statements from linear algebra that are useful for a better understanding of the geometry and properties of residual subspaces. First, we define the notion of *explained variance* and *relative explained variance* to be able to quantify the properties of residual subspaces.

Definition 1 (Subspace explained variance (Shen and Huang, 2008; Gandelsman et al., 2023))**.** *Let* $\mathcal{D} \subset \mathbb{R}^d$, $\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ *be a dataset, and* $S \subset \mathbb{R}^d$ *is an arbitrary subspace, with* $Pr(\mathbf{x})$:

$\mathbb{R}^d \rightarrow S$ *being the projection function onto* S*. We call the variance of the projections* $Pr(\mathcal{D})$ *the* explained variance *of subspace* S *with respect to* $\mathcal{D}$*:*

$$\mathrm{EV}^{\mathcal{D}}(S) = \mathbb{E}_{\mathcal{D}}\|\mathrm{Pr}(\mathbf{x} - \mathbb{E}[\mathbf{x}])\|^2 = \\ = \frac{1}{N}\sum_{\mathbf{x}\in\mathcal{D}} \|\mathrm{Pr}(\mathbf{x}) - \mathrm{Pr}(\boldsymbol{\mu})\|^2,$$

where $\boldsymbol{\mu} = \frac{1}{N}\sum_{\mathbf{x}\in\mathcal{D}} \mathbf{x}$*.*

If $\bar{X}$ is a matrix of centered data vectors $(\mathbf{x} - \boldsymbol{\mu})$ for $\mathbf{x} \in \mathcal{D}$ (row-wise), and V is the $k \times d$ matrix defining an arbitrary basis of the subspace S, $S = \langle v_1, \dots, v_k \rangle$, then the explained variance $\mathcal{EV}^{\mathcal{D}}(S)$ can be written in matrix form:

$$\mathrm{EV}^{\mathcal{D}}(S) = \mathrm{Tr}(\mathrm{Pr}(\bar{X})^T \mathrm{Pr}(\bar{X})), \qquad (4)$$

where the projection operator $\mathrm{Pr}(X)$ can be computed as

$$\mathrm{Pr}(\bar{X}) = \bar{X}V^T(VV^T)^{-1}V. \qquad (5)$$

In the case of an orthonormal basis, $V^TV = \mathbb{I}$, formulas (4) and (5) become a simple decomposition into the sum of component-wise variations:

$$\mathrm{EV}^{\mathcal{D}}(S) = \sum_{i=1}^{k} \mathcal{V}_i^{\mathcal{D}}, \qquad (6)$$

where $\mathcal{V}_i^{\mathcal{D}}$ is the variance along the ith basis vector.

Relative explained variance reflects the relative importance of a subspace by the ratio of the subspace explained and total variance of the data:

$$\mathrm{RV}^{\mathcal{D}}(S) = \frac{\mathrm{EV}^{\mathcal{D}}(S)}{\mathrm{Var}(\mathcal{D})}.$$

For data distributed equally over all directions, it is proportional to the subspace dimension. For example, for $\mathcal{D} \sim \mathcal{N}(\mu, \sigma^2)$ for any subspace S

$$\mathrm{RV}^{\mathcal{D}}(S) = \frac{\dim(S)}{d}.$$

Definition 2. *A subspace* S *is called an* α-residual subspace *with respect to* $\mathcal{D}$ *if and only if its relative explained variance is not greater than* α*:*

$$\mathrm{RV}^{\mathcal{D}}(S) \leq \alpha. \qquad (7)$$

The simplest way to find residual subspaces for a given α follows from (6). We can compute the variances Var_i with respect to each coordinate of the embeddings, and then select the coordinates $\{u_{i_1}, \dots, u_{i_m}\}$ with the smallest variances while their sum does not exceed the desired portion of the total variance. But this method does not guarantee that the required subspace will be found even if it exists for a given dataset. Figure 5 shows the geometric intuition of our approaches; in particular, the residual subspace, even if it exists, may not be spanned by any subset of the standard basis. The following proposition provides a guaranteed way to find the α-residual subspace if it exists.

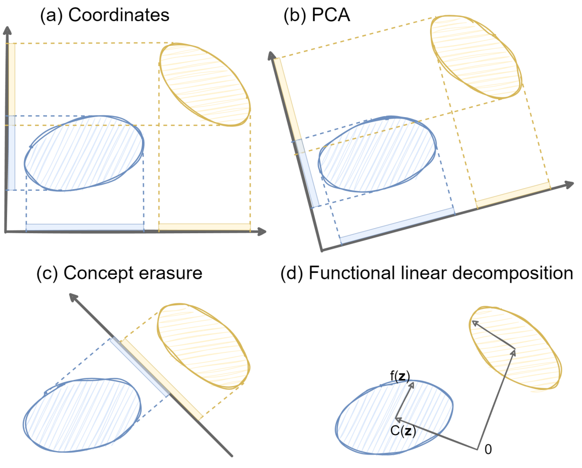

Figure 5: Geometric intuition of our approaches.

Proposition 1. *Let* $\{\mathbf{u}_1, \dots, \mathbf{u}_d\}$ *be the principal components of a dataset* $\mathcal{D}$ *with corresponding singular values* $\lambda_1, \dots, \lambda_d$ *(in descending order). Then the explained variance of a subspace spanned by* $d - k$ *last principal components* $R_k = \langle \mathbf{u}_{k+1}, \dots, \mathbf{u}_d \rangle$ *is*

$$\mathrm{EV}^{\mathcal{D}}(R_k) = \sum_{i=k+1}^{d} \lambda_i. \qquad (8)$$

Moreover, R_k *has the minimal explained variance among all* $(d-k)$*-dimensional subspaces.*

Proof. The first statement follows from (4), taking in account that the trace of a matrix is invariant under the change of the basis. Therefore, we can apply a singular transform to $\bar{X}$ and obtain

$$\mathrm{Tr}(\mathrm{Pr}_i(\bar{X})^T \mathrm{Pr}_i(\bar{X})) = \\ = \mathrm{Tr}(\mathrm{Pr}_i(\mathrm{diag}(\lambda_1, \dots, \lambda_d))) = \lambda_i.$$

The second statement follows from the Frobenius theorem, which says that for any matrix $\bar{X}$ the projection of its rows to the first k singular components leads to the best rank-k approximation with respect to Frobenius norm:

$$\langle \mathbf{u}_1, \dots, \mathbf{u}_k \rangle = \underset{S, \dim S = k}{\operatorname{argmin}} \sum_{\mathbf{x} \in \bar{X}} \|\mathbf{x} - \operatorname{Pr}_S(\mathbf{x})\|^2,$$

where the sum goes over rows of $\bar{X}$. This can be rewritten in terms of the residual subspace $R = \langle \mathbf{u}_{k+1}, \dots, \mathbf{u}_d \rangle$, which is unambiguously defined as the orthogonal complement of $S = \langle \mathbf{u}_1, \dots, \mathbf{u}_k \rangle$:

$$\begin{aligned} \langle \mathbf{u}_{k+1}, \dots, \mathbf{u}_d \rangle &= \underset{R, \dim R = d-k}{\operatorname{argmin}} \sum_{\mathbf{x} \in \bar{X}} \|\operatorname{Pr}_R(\mathbf{x})\|^2 \\ &= \underset{R, \dim R = d-k}{\operatorname{argmin}} \operatorname{EV}^{\mathcal{D}}(R), \end{aligned}$$

which completes the proof. □

As a corollary, PCA allows to find the α-residual subspace for a given dataset $\mathcal{D}$, if it exists. Namely, we can select its singular values starting from the least until their relative sum exceeds α. Then, the number of components in the sum is equal to the maximal subspace dimension, and the subspace spanned by the corresponding singular vectors provides the necessary subspace.

A.2 Head-wise decomposition

In our derivation of the form of head-wise flows, we follow the ideas proposed by Gandelsman et al. (2023). In the following, we consider Transformer blocks with post-layer-normalization, such as in BERT and RoBERTa models. The transformation inside each layer can be written as

$$\hat{\mathbf{z}}_l = \operatorname{LN}(\mathbf{z}_{l-1} + \operatorname{MHA}(\mathbf{z}_{l-1})), \quad (9)$$

$$\mathbf{z}_l = \operatorname{LN}(\hat{\mathbf{z}}_l + \operatorname{MLP}(\hat{\mathbf{z}}_l)), \quad \text{where} \quad (10)$$

$$\operatorname{LN}(\mathbf{x}) = \frac{\mathbf{x} - \bar{\mathbf{x}}}{\|\mathbf{x} - \bar{\mathbf{x}}\|^2}, \quad (11)$$

and $\bar{\mathbf{x}} = \frac{1}{d} \sum_{i=1}^d x_i$ is the mean of the components of a vector $\mathbf{x}$. The numerator of (11) can be rewritten as a linear transform

$$\mathbf{x} - \bar{\mathbf{x}} = (\mathbf{I} - \frac{1}{d}\mathbf{1})\mathbf{x} = \Pi\mathbf{x}, \quad (12)$$

where $\mathbf{I}$ is the identity matrix, $\mathbf{1}$ is the square matrix consisting of ones, and d is the dimension of $\mathbf{x}$. Note that this transform is in fact an orthogonal projection to the hyperplane defined by the equation $x_1 + \dots + x_d = 0$. As all projections, Π is idempotent:

$$\Pi^2 = \Pi. \quad (13)$$

Applying (12) and (13) to (9), we can write a layer-wise linear decomposition for post-layer-norm Transformers:

$$\begin{aligned} M(\mathbf{z}) = \alpha(\mathbf{z})\Pi(\mathbf{z}_0) + \sum_l \beta_l(\mathbf{z})\Pi(\operatorname{MLP}(\hat{\mathbf{z}}_l)) + \\ + \sum_l \gamma_l(\mathbf{z})\Pi(\operatorname{MHA}(\mathbf{z}_{l-1})) = \\ = \alpha(\mathbf{z})\Pi(\mathbf{z}_0) + \sum_l \beta_l(\mathbf{z})\Pi(\operatorname{MLP}(\hat{\mathbf{z}}_l)) + \\ + \sum_l \sum_h \gamma_l(\mathbf{z})\Pi(A^{l,h}(\mathbf{z}_{l-1})), \quad (14) \end{aligned}$$

where α, β, γ are input-dependent scalars, Π is the projection transform (12), and $A^{l,h}$ denotes attention head h on layer l.

B Technical details of the experiments

B.1 Preprocessing and models

For text preprocessing, we only replaced consecutive spaces, trailing spaces, and a newline characters with one space, as was done by Tulchinskii et al. (2023).

For embeddings extraction, we used standard pretrained models from the HuggingFace[4] library: `roberta-base` (125M parameters), `microsoft/phi-2` (2.7B parameters), `bert-base-uncased` (110M parameters). We use each text sample as an input for chosen model and obtain the resulting embedding from the last layer of this model. We take the mean pooling of that embedding to decrease the dimensionality and get a vector of dimension 768; this is our text feature vector.

For all further experiments with embeddings, we use the logistic regression model from the *scikit-learn*[5] package on the training subset with default parameters: *lbfgs* solver, L_2 regularization coefficient $C = 1$, and maximum amount of iterations $\text{max\_iter} = 100$.

B.2 Computational resources

For all of our experiments we used two servers with the following computational resources:

- 1 V100 16Gb GPU + 32 CPUs (Intel(R) Xeon(R) Gold 6151), 126GB RAM
- 2 V100 16GB GPUs + 64 CPUs (Intel(R) Xeon(R) Gold 6151), 252GB RAM

[4] https://huggingface.co/

[5] https://scikit-learn.org/stable/

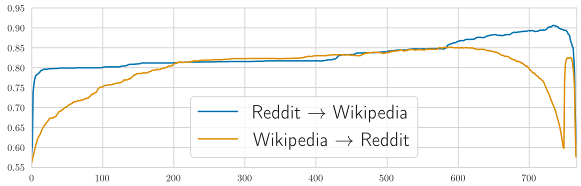

Figure 6: Accuracy (vertical axis) as a function of the number of components removed from the RoBERTa embedding (horizontal axis).

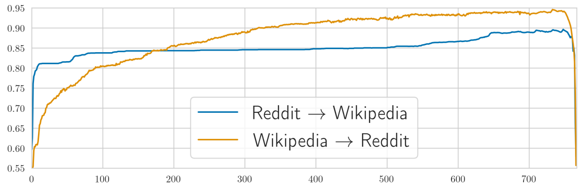

Figure 7: Accuracy (vertical axis) as a function of the number of removed components (similar to Fig. 6) for data with all symbols except English letters, numbers, and "!", "?", ",", and "." symbols filtered out.

B.3 Detailed experimental setup on GPT-3D

For experiments on the GPT-3D dataset, we consider texts generated by either the *davinci* or *GPT-4-o* generator on the ith topic from the list and the corresponding human-written texts on the same topic as one dataset, labeling the generated and human-written texts with "0" and "1" respectively. We use each text sample as an input for the RoBERTa model and take the mean-pooled embeddings to obtain a vector of dimension 768; this is our text feature vector.

We split the resulting dataset of these feature vectors into training and test subsets. We train logistic regression on the training subset and test the resulting classifier on the test subset of every other generator we have. The resulting accuracy values comprise the ith row of our resulting diagram. We repeat this process for every considered topic.

B.4 Greedy search for embedding components

The resulting scores for greedy search of the embeddings components to remove, in both directions, $S_{\text{Reddit}\to\text{Wikipedia}}$ and $S_{\text{Wikipedia}\to\text{Reddit}}$, are shown in Figure 6. We also provide another similar plot in Figure 7 in the setting where all symbols except English letters, numbers, and "!", "?", ",", and "." symbols have been filtered out. This experiment shows that the text preprocessing method can significantly influence the process of choosing the best components.

B.5 Layer-wise head pruning on GPT-3D dataset, exrtended with GPT-4 generations

The GPT-3D dataset contains natural and artificially generated texts (by two models: GPT-3.5-davinci-003 and GPT-4-o) in three different domains: *Wikipedia* articles, long-form question answering from *Reddit* (general topics), and *Stack-Exchange* (more technical texts). For each (domain, generating model) pair, the dataset contains an equal number of generated and natural texts from that domain; therefore, classes are balanced in all settings. For each (domain, generating model) pair, we split the data into training and evaluation subsets in the 13:2 ratio. None of the evaluation subsets intersect with any of the training subsets.

Although our main track of research on our GPT-3D dataset was conducted using GPT-4-o data, we also generated a small sample of data by the earlier GPT-4 generator. This model is more expensive so the amount of data fit to our budget was not sufficient for a stable evaluation of all the proposed methods; but below we report interesting findings obtained by layer-wise head pruning. Table 5 demonstrates, that in this data-sparse regime the performance of OOD transfer of GPT-4 generations is low, but 1st layer pruning corrects it by as much as 16%. This observation does not correspond to the results obtained by GPT-4-o generations. Besides, the quality of cross-model transfer significantly improved. We believe that this observation requires an additional study with a larger GPT-4 dataset.

Below we described the detailed experimental setup for this study.

The experiment was conducted as follows: first, a classifier was trained on data for one (domain, generating model) pair and then evaluated on two other domains with the same generating model; we call this the OOD (out-of-domain) setting. Then, the classifier is evaluated on all three domains but with a different generating model (Transfer). The results are presented in Table 5, which reports average accuracy across all domains.

The first row of the table (Full) contains results obtained using the unaltered RoBERTa-base model. Then we separately prune each layer of attention heads ("turn off" all 12 attention heads of each layer by zeroing their output); this can be done, e.g., with the `prune_heads` method of the RoBERTaModel class from the HuggingFace library. Re-

	davinci OOD	GPT-4 OOD	davinci to GPT-4 transfer	GPT-4 to davinci transfer
Full model	81.3	64.3	66.4	70.4
Pruned layer				
#0	83.2	**80.1**	**80.0**	**83.2**
#1	83.4	78.8	74.7	79.2
#2	82.1	78.8	72.7	77.4
#3	81.8	82.0	73.6	78.1
#4	83.4	79.6	71.6	76.4
#5	82.2	78.1	72.9	75.6
#6	84.0	76.3	72.3	74.4
#7	82.8	75.4	70.1	74.6
#8	82.8	72.1	68.5	73.4
#9	83.2	73.2	68.7	71.4
#10	83.1	71.0	68.1	72.8
#11	**86.6**	68.2	67.3	71.7

Table 5: Average accuracy of artificial text detection over three domains (Wikipedia, Reddit, StackExchange) and two generating models (GPT3.5-davinci and GPT4). Detector is trained on one domain against one generator and evaluated on other domains (OOD) and on all domains against unseen generating model (transfer). Best results are given in bold, runner-ups are underlind.

	Cross-domain			Cross-model		
	Avg	Max ↑	Max ↓	Avg	Max ↑	Max ↓
Roberta	73.0	-	-	82.8	-	-
Bshift	73.0	6.4	-6.8	82.2	1.5	-2.6
CoordInv	72.1	1.1	-3.7	82.1	0.9	-3.4
ObjNum	72.9	0.9	-1.5	83.0	0.7	**-0.0**
SOMO	72.9	6.8	-3.8	82.1	0.6	-4.1
Tense	72.7	0.4	-1.6	82.8	1.0	-0.4
SentLen	73.0	4.1	-3.0	82.6	0.2	-1.2
SubjNum	72.8	0.4	-1.6	82.9	0.5	-0.4
TopConst	**75.1**	**12.6**	-1.8	**83.1**	2.2	-0.9
TreeDepth	73.9	12.1	**-1.4**	83.0	1.0	-0.3
WC	74.1	11.0	-8.5	83.0	**2.9**	-2.9

Table 6: Balanced accuracy results for out-of-domain classification for different erased concepts on SemEval

sults for these cases are reported in other rows of Table 5.

Table 5 shows full results across the layers, indicating that pruning the lower layers of the model, especially Layer 0, yields better results.

B.6 Concept erasure on SemEval

Table 6 reports detailed results on concept erasure on the SemEval dataset. For concept erasure we use an open-source implementation[6].

C Cross-dataset transfer

Table 7 compares the classifiers trained on SemEval dataset with the same setup trained on GPT-3D data, but tested on SemEval. Surprisingly, in cross-domain transfer heads and coordinates selection on GPT-3D leads to an improvement of the performance on SemEval. However, the cross-model performance degrades.

	SemEval		**GPT-3D**		
RoBERTa	CD	CM	CD	CM	CA
Baseline	73.0 / 76.4*	82.8 / 76.3*	84.1	71.0	70.1
Selected heads	74.3 / 75.6*	80.0 / 75.4*	86.6	79.3	79.2
Selected coordinates	74.5 / 75.4*	82.6 / 75.3*	85.4	71.9	72.8

Table 7: Balanced accuracy for OOD classification: cross-domain (CD), cross-model (CM), cross-all (CA). Numbers with asterisks correspond to cross-dataset transfer.

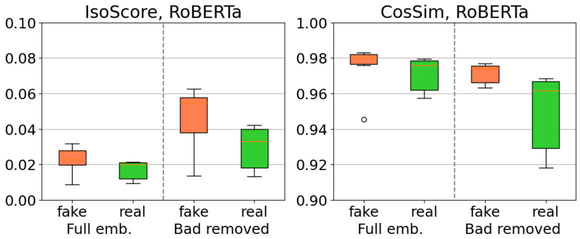

Figure 8: IsoScore and cosine similarity of the RoBERTa embeddings before and after removing their "bad" components; the embeddings were calculated on GPT-3D dataset.

D Removing "bad" outliers and how it influences the geometry of embeddings

Previous studies have shown that some dimensions skew the embedding space greatly and have a dramatic influence on its geometry. In particular, Timkey and van Schijndel (2021) have shown that the embeddings of BERT, RoBERTa, and some other Transformer-based models lie in a narrow cone. To show this, they use the mean cosine similarity of the embeddings: if the cosine similarity of all embeddings is high, it means that they are similar to each other along some dimensions; the larger the average cosine similarity, the less isotropic the embedding space is.

Rudman et al. (2022) introduced a more complex tool for measuring the anisotropy of the embedding space: IsoScore. The fundamental motivation for IsoScore is that it roughly reflects the fraction of dimensions uniformly utilized by a given point cloud. According to the authors' estimation, less

[6] https://github.com/EleutherAI/concept-erasure

than 20% of dimensions of the BERT model embedding space are utilized uniformly. Larger IsoScore values correspond to more isotropic embedding spaces.

Figure 8 shows how removing the components that are "bad" for cross-domain and cross-model generalization abilities influences the IsoScore and cosine similarity scores for RoBERTa embeddings.

We see that after removing these "bad" dimensions, the embeddings of fake and real texts change their isotropy in different rates, but both become more isotropic in general. Based on this observation, we hypothesize that the isotropy of the embedding space can be connected to the model's generalization abilities; we leave testing this hypothesis for future research.

E Components removal and PHD

We conducted additional experiments to evaluate the influence of removing embedding components (selected with the greedy search outlined in Section 3.2 Subspace removing methods) in the RoBERTa and BERT models on the cross-domain and cross-model generalization abilities of the persistent homological fractal intrinsic dimensionality-based method. Figure 9 shows a consistent decrease in accuracy for both cross-model and cross-domain ATD as components are being removed. Such removal typically reduces the intrinsic dimensionality of human-written texts, hence degrading the discriminative power of linear classifiers in ATD.

An interesting observation is that the PHD of a newer generation LLM (GPT-4o) is higher than that of human-written texts, while the PHD of the older generation (GPT-3.5-davinci) is lower that of human-written texts. This may explain the poor generalization ability between the models on GPT-3.5-davinci and GPT-4o. See Figure 10 for details.

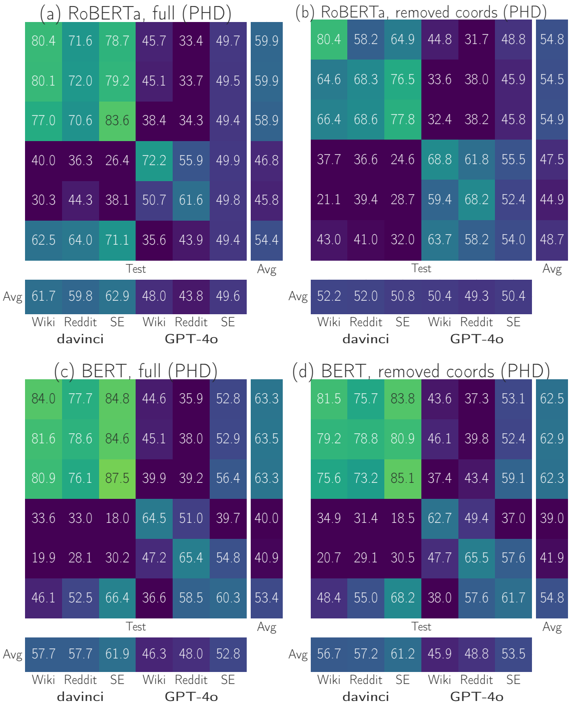

Figure 9: PHD-based logistic regression accuracy before and after components removal, mean accuracy in cross-domain/cross-model ATD on GPT-3D: (a) RoBERTa, full embeddings, (b) RoBERTa after components removal, (c) BERT, full embeddings, (d) BERT, after components removal.

F PCA

We investigated the PCA decomposition of the embedding spaces of RoBERTa, BERT and Phi-2. We tried to remove components with highest and lowest variance to check how it affects the overall accuracy and generalization abilities of the models. The results are shown in Figures 13 and 14.

Figure 13 shows that while we remove PCA components of the RoBERTa embedding space with the largest variance, the transferability between the different domains and models drops significantly. At first, the transferability from GPT-4o to GPT-3.5-davinci goes down to random; next, transferability between different domains of texts generated with GPT-3.5-davinci goes down to random; and finally, transferability between GPT-3.5-davinci and GPT-4o drops down. Interestingly, transferability between different domains of GPT-4o remains significantly higher than random even after removing 90% of the high-variance components.

Figure 14 shows that removing the first PCA component with the highest accuracy does not affect the classification quality much, suggesting that it does not play a distinct role in classification. However, removing 25% of the components with high variance is damaging for all three models, while removing the components with low or average variance does not hurt the model performance.

Overall, we see that high-variance components in the PCA space generally play some important role in the generalization ability of all three models (RoBERTa, BERT, and Phi-2); however, we have not been able to significantly improve the quality of classification by simply removing low-variance PCA components on any model.

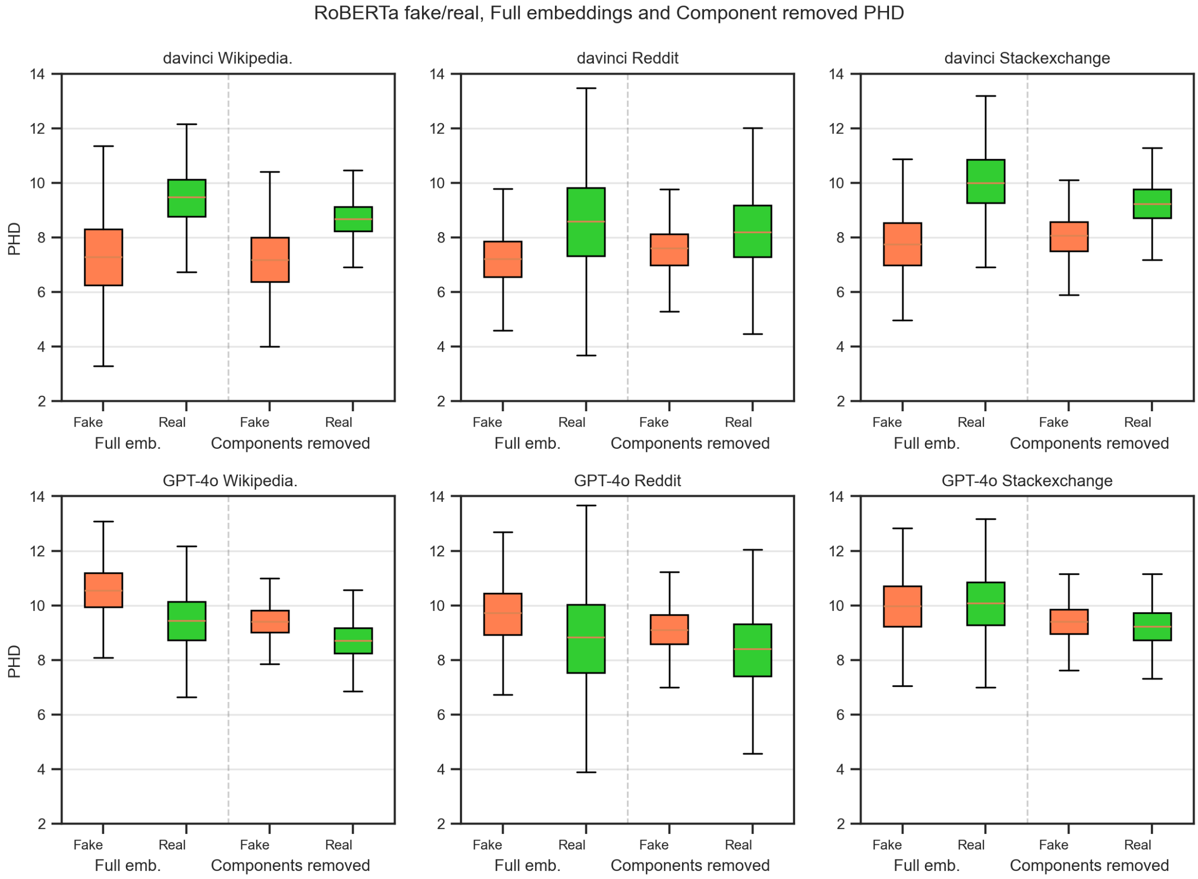

Figure 10: PHD of RoBERTa full embeddings and embeddings after component removal for real/fake texts from the GPT-3D dataset.

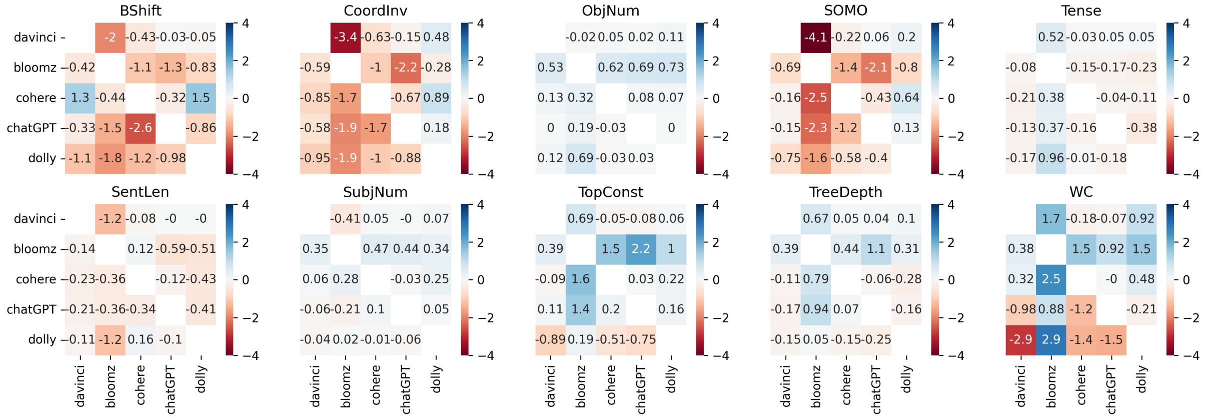

Figure 11: Concept erasure, cross-model setting

G Datasets license

We release our dataset under CC BY-SA 4.0 licence agreement. For the information about the licence of M4 (SemEval) subsets, see original paper by Wang et al. (2024b).

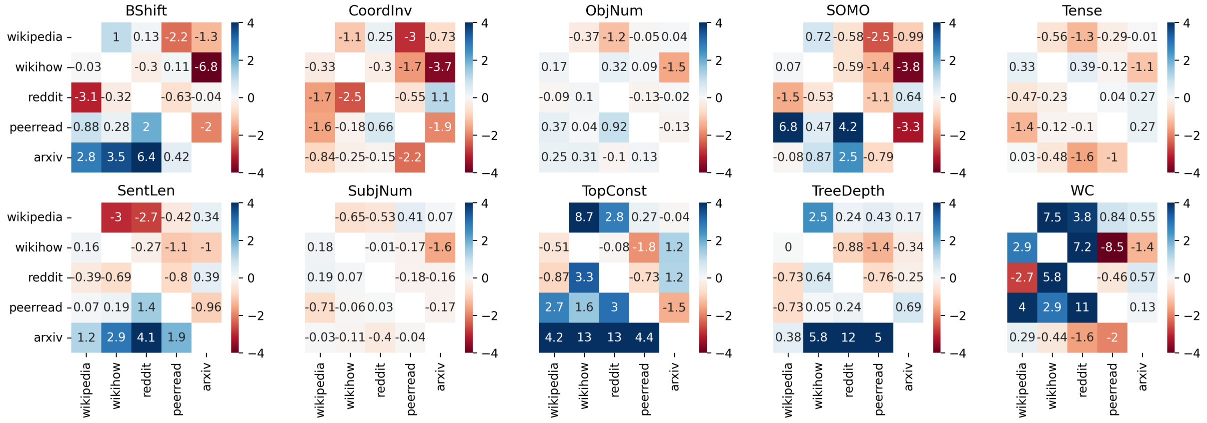

Figure 12: Concept erasure, cross-domain setting

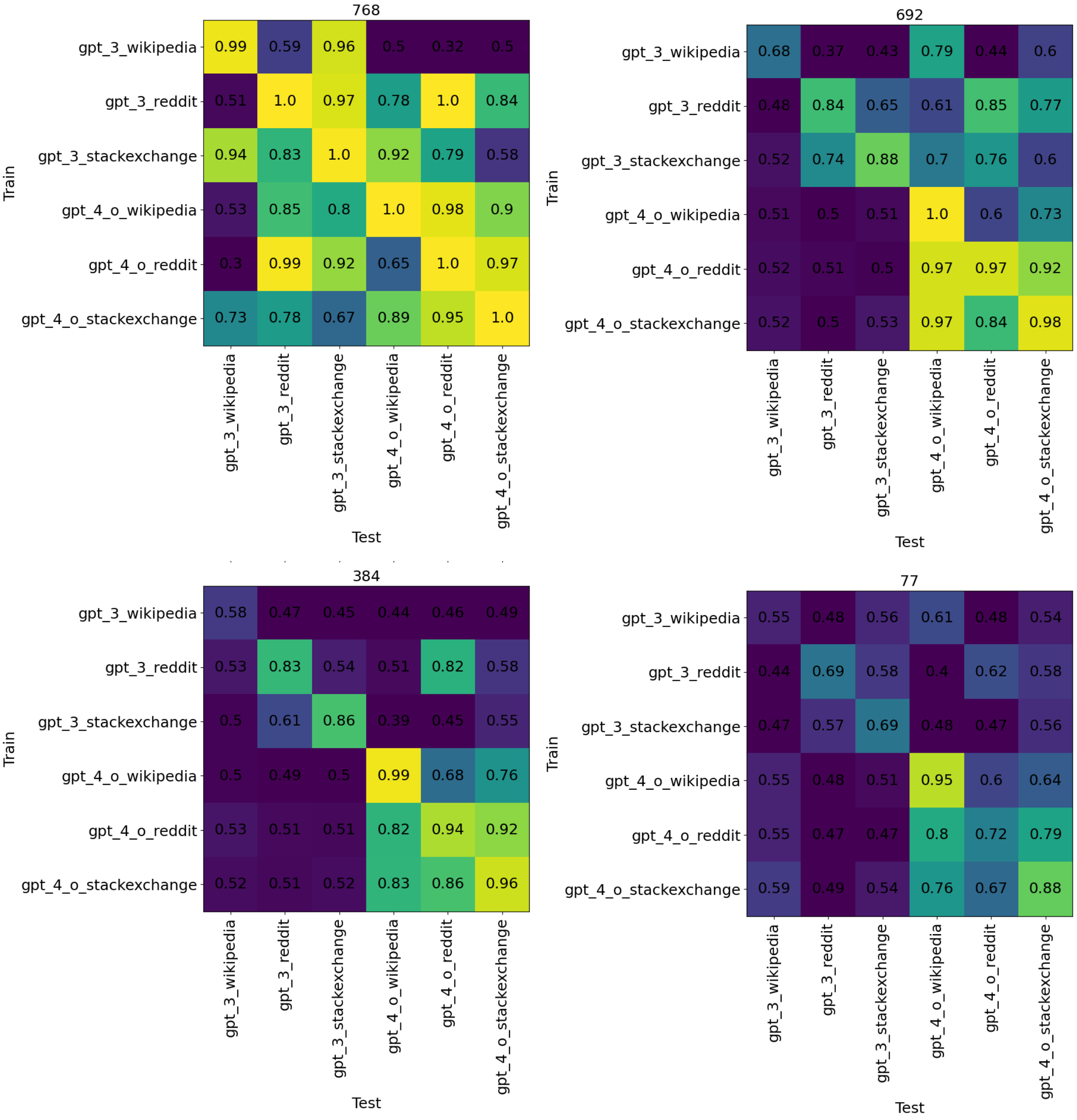

Figure 13: Classification quality on PCA components of RoBERTa embeddings on the GPT-3D dataset. Top left — all components are present; top right — 10% of the components with the largest variance are removed; bottom left — 50% of the components with the largest variance are removed; bottom right — 90% of the components with the largest variance are removed.

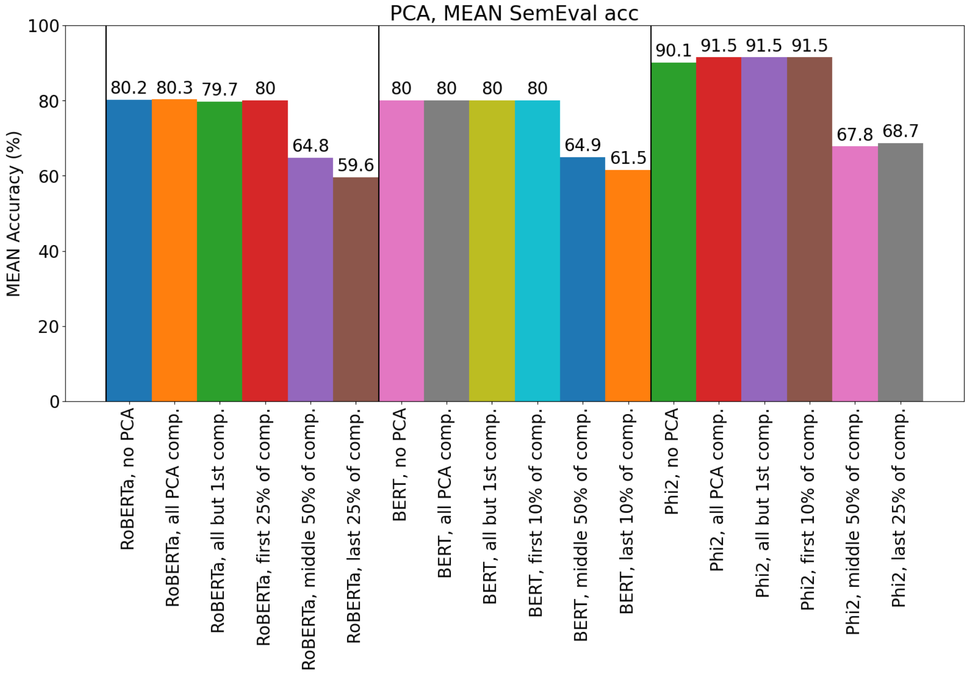

Figure 14: Mean accuracy on the GPT-3D dataset, depending on the number of PCA components left; e.g., "top 10% components" means that we have removed 90% of the components with the smallest variance.