Comparing Genomic LLM Variant Pathogenicity Predictions with Established In-Silico Scores, 2025, 1–12
doi:

Comparing Genomic LLM Variant Pathogenicity Predictions with Established In-Silico Scores

Reena Assassa,[1] Lillianna Gund[2] and Shaun Ku[3]

[1]Department of Computer Science, Johns Hopkins University, 3400 N. Charles St., Baltimore, MD, 21218, USA, [2]Department of Computer Science, Johns Hopkins University, 3400 N. Charles St., Baltimore, MD, 21218, USA and [3]Department of Computer Science, Johns Hopkins University, 3400 N. Charles St., Baltimore, MD, 21218, USA

*Email: rassass1@jhu.edu, lgund1@jhu.edu, sku5@jhu.edu

Abstract

Recent advances in genomic large language models (LLMs) have opened new possibilities for predicting the pathogenicity of genetic variants directly from sequence context. In this project, we compare the performance of fine-tuned genomic LLMs such as DNABERT and DT-GPT on ClinVar variant-context windows against established in-silico pathogenicity predictors including FATHMM-MKL, SIFT, and PolyPhen-2. Our approach involves fine-tuning DNABERT and DT-GPT on 101-base-pair genomic windows centered on variants extracted from the ClinVar GRCh38 dataset, while using identical variant sets to obtain predictions from existing tools. We have prepared a balanced dataset of approximately 33,000 variants, with an 80/20 train–test split stratified by gene to minimize bias across functional groups. The goal is to evaluate each model's predictive accuracy, calibration, and interpretability, as well as to identify cases where contextual learning by LLMs offers complementary or inferior insights relative to traditional feature-based models. We find that while LLMs can learn non-trivial decision boundaries under balanced training, established in-silico tools still provide more reliable performance on this dataset.

Key words: LLMs, DNABERT, DT-GPT, FATHMM-MKL, SIFT, PolyPhen-2

1. Introduction

1.1. Background

Advances in large language models (LLMs) have transformed natural language processing and are now being applied to biological sequences, treating DNA as a "language" with its own syntax and semantics. Models such as DNABERT (1) and DT-GPT (2) leverage transformer architectures to learn contextual relationships within nucleotide sequences, enabling them to identify biologically meaningful motifs and variant effects.

In parallel, established in-silico pathogenicity predictors such as FATHMM-MKL (7), SIFT (6), and PolyPhen-2 have long been used in clinical genomics to assess variant deleteriousness by integrating diverse annotation features and evolutionary conservation scores. However, these traditional models often rely on predefined statistical features and may underperform on variants in poorly annotated or non-coding regions.

1.2. Goal

This project aims to fine-tune genomic LLMs on ClinVar variant-context sequences and benchmark their predictive performance against conventional tools. By comparing their outputs across shared variant datasets, we aim to explore whether LLMs can capture richer contextual signals that enhance pathogenicity prediction and reveal cases where language-based models complement or surpass traditional scoring methods.

1.3. Data Collection and Processing

To construct a dataset of clinically validated variants, we retrieved records from the ClinVar database (GRCh38 release) corresponding to several well-known and well-characterized single-gene disorders: Adenosine Deaminase (ADA) Deficiency, Alpha-1 Antitrypsin Deficiency (Alpha-1), Cystic Fibrosis (CF), Galactosemia, Maple Syrup Urine Disease (MSUD), Neurofibromatosis Type 1 (NF1), Phenylketonuria (PKU), Severe Combined Immunodeficiency (SCID), Sickle Cell Disease/Anemia, and Smith-Lemli-Opitz Syndrome (SLOS). The downloaded ClinVar files contained fields such as variant name, associated

gene(s), protein change, condition(s), accession identifiers, and genomic coordinates.

We created a custom Python script utilizing the Ensembl REST API to translate each ClinVar entry into a genomic sequence formatted for DNABERT and other sequence-based models. For each variant, the script retrieved 200 bp of genomic context surrounding the variant position and extracted a 101 bp window centered on the site for input to DNABERT. Each variant was labeled according to its clinical classification: `Pathogenic` and `Likely pathogenic` were encoded as `1`, while `Benign` and `Likely benign` were encoded as `0`. Variants with uncertain or conflicting significance were excluded. The final aggregated dataset included the gene name, SPDI notation, ClinVar classification, binary label, extracted sequence, and gene group.

To ensure balanced representation across genes and disease types, we implemented a second Python script to randomly split the dataset into training and test sets using a stratified bucket method. This process produced approximately 33,000 total variants, with an 80/20 train–test ratio stratified by gene. Splitting by gene reduces information leakage between training and testing while maintaining proportional representation across variant categories.

Gene_group	Total	Train	Test	Train_%	Test_%
ADA	910	728	182	80.0	20.0
ADA|LOC107303343	230	184	46	80.0	20.0
BCKDHA	719	575	144	80.0	20.0
BCKDHB	713	570	143	79.9	20.1
CD3D	110	88	22	80.0	20.0
CD3G	34	27	7	79.4	20.6
CD3G|LOC126861358	32	25	7	78.1	21.9
CEP290	3056	2444	612	80.0	20.0
CEP290|LOC129390514	30	24	6	80.0	20.0
CEP290|RLIG1	73	58	15	79.5	20.5
CEP83	22	17	5	77.3	22.7
CFTR	3156	2524	632	80.0	20.0
CFTR-AS1|CFTR	54	43	11	79.6	20.4
CFTR-AS2|CFTR	32	25	7	78.1	21.9
CFTR|CFTR-AS1	466	372	94	79.8	20.2
CFTR|CFTR-AS2	602	481	121	79.9	20.1
CFTR|CFTR-AS2|LOC111674472	371	296	75	79.8	20.2
CFTR|LOC111674463	49	39	10	79.6	20.4
CFTR|LOC111674475	122	97	25	79.5	20.5
CFTR|LOC111674477	159	127	32	79.9	20.1
CFTR|LOC113664106	35	28	7	80.0	20.0
CIITA	1010	808	202	80.0	20.0
COL1A1	124	99	25	79.8	20.2
COL2A1	55	44	11	80.0	20.0
COL3A1	27	21	6	77.8	22.2
COL7A1	34	27	7	79.4	20.6
DBT	733	586	147	79.9	20.1
DCLRE1C	568	454	114	79.9	20.1
DHCR7	540	432	108	80.0	20.0
FOXN1	27	21	6	77.8	22.2
GALE	354	283	71	79.9	20.1
GALK1	498	398	100	79.9	20.1
GALT	731	584	147	79.9	20.1
GALT|LOC130001683	83	66	17	79.5	20.5
GLIS2	149	119	30	79.9	20.1
HBB|LOC106099062|LOC107133510	108	86	22	79.6	20.4
HBB|LOC107133510|LOC110006319	42	33	9	78.6	21.4
IL2RG	259	207	52	79.9	20.1
IL2RG|LOC126863274	27	21	6	77.8	22.2
IL7R	298	238	60	79.9	20.1
INVS	851	680	171	79.9	20.1
IQCB1	387	309	78	79.8	20.2
JAK3	749	599	150	80.0	20.0
LIG4	32	25	7	78.1	21.9
LOC102724058|SCN1A	71	56	15	78.9	21.1
LOC111674477|CFTR	21	16	5	76.2	23.8
LOC111811965|MIR4733HG|NF1	51	40	11	78.4	21.6
LOC126861615|PAH	53	42	11	79.2	20.8
LOC129937586|NPHP3|NPHP3-ACAD11|NPHP3-AS1	63	50	13	79.4	20.6
NEK8	21	16	5	76.2	23.8
NF1	4714	3771	943	80.0	20.0
NF2	735	588	147	80.0	20.0
NHEJ1	78	62	16	79.5	20.5
NPHP1	701	560	141	79.9	20.1
NPHP3-ACAD11|NPHP3	57	45	12	78.9	21.1
NPHP3|NPHP3-ACAD11	743	594	149	79.9	20.1
NPHP3|NPHP3-ACAD11|NPHP3-AS1	40	32	8	80.0	20.0
NPHP4	1427	1141	286	80.0	20.0
PAH	971	776	195	79.9	20.1
PPM1K	76	60	16	78.9	21.1
PTPRC	683	546	137	79.9	20.1
RAG1	419	335	84	80.0	20.0
RAG2	259	207	52	79.9	20.1
RFX5	141	112	29	79.4	20.6
RFXANK	134	107	27	79.9	20.1
RFXAP	35	28	7	80.0	20.0
SCN1A	87	69	18	79.3	20.7
SCNN1B	191	152	39	79.6	20.4
SERPINA1	414	331	83	80.0	20.0
SPRED1	267	213	54	79.8	20.2
TTC21B	916	732	184	79.9	20.1
TTC21B|TTC21B-AS1	95	76	19	80.0	20.0
Unknown	934	747	187	80.0	20.0
WDR19	21	16	5	76.2	23.8

Fig. 1: Snippet of terminal output of trained test split, utilizing bucketed split to preserve proportional class representation.

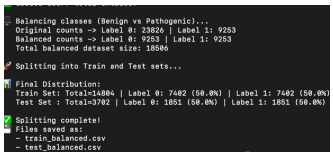

Fig. 2: Snippet of terminal output of trained test split which enforced strict balanced between the classes of labels

Later on, we found that due to the utilized bucket method, that there was an uneven portion of data labeled pathogenic vs. benign. We were able to re-purpose the python script to instead force balance between the 2 labels.

2. Methods

2.1. Model Configuration

We will fine-tune DNABERT-k (with $k = 6$ for k-mer tokenization) using pretrained HuggingFace weights. Training will be performed on the Johns Hopkins Rockfish cluster. The configuration should include multiple epochs, a learning rate in the range of 1e-5 to 5e-5, and batch sizes of 16–32, depending on sequence length and GPU memory availability. We will also explore adaptation of DT-GPT for comparative purposes as a domain-tuned generative model for genomic context modeling (2).

2.1.1. DT-GPT

We fine-tuned DT-GPT (a Mistral-7B–based genomic language model) using LoRA lightweight adaptation. All models were trained on 101-bp ClinVar variant-centered windows.

Unbalanced Fine-Tuning (3 epochs).
Our initial run used the original ClinVar-derived dataset, which may be skewed toward benign variants. After three epochs of LoRA fine-tuning on Rockfish, the model consistently predicted all variants as pathogenic, indicating a collapsed decision boundary.

Unbalanced Without ClinVar Label Text (5 epochs).
To reduce label leakage, we removed all classification phrases (e.g., “pathogenic,” “likely benign”) from prompts and retrained for five epochs. Despite these adjustments, the model again converged to predicting the pathogenic class exclusively.

Balanced Dataset Fine-Tuning (intended 30 to 50 epochs).
We constructed a fully balanced training/testing split to eliminate class imbalance. A long-run LoRA fine-tune was launched on Rockfish, but the job stopped after partial progress due to the course wide GPU quota being exhausted. The interrupted checkpoint did not contain complete LoRA weights and could not be evaluated.

One-Epoch Balanced Fine-Tune on Google Colab.
To quickly obtain a complete model, we reproduced training on Google Colab with an T4 GPU. A full one epoch balanced-dataset LoRA fine-tune completed successfully, producing a usable DT-GPT adapter for evaluation.

Evaluation.
All models were evaluated via a custom prediction pipeline that (1) loaded the base Mistral model and merged LoRA weights, (2) generated free-form predictions, (3) mapped text responses

to binary labels, and (4) computed accuracy, precision, recall, F1-score, and confusion matrices. Evaluation was performed on CPU due to cluster GPU limits.

2.2. DNABERT-2 Fine-Tuning Procedures

We were able to fine-tune the publicly available DNABERT2 using HuggingFace to perform binary classification on our ClinVar dataset. Unlike the cloud-based training used for DT-GPT, this model was trained locally.

Local Implementation and Hardware Constraints.
All DNABERT-2 experiments were conducted on a local personal computer running Windows 11 and utilizing an NVDIA RTX 3080 GPU. To accommodate the Windows OS environment, the model architecture was downloaded locally, and Linux-specific dependencies, mainly Triton compiler, were removed. This allowed for direct utilization of the GPU's CUDA cores without virtualization or the need for the remote rockfish cluster.

To mitigate risks of interruptions during extended training runs, of which could last several hours, we implemented a checkpointing strategy. The training pipeline was configured to serialize the model state at regular intervals, allowing for the experiment to resume seamlessly for the most recent step in event of a crash. For example our 200-epoch experiment yielded multiple intermediate retention points (e.g., `checkpoint-111200` and `checkpoint-594800`), ensuring data continuity across sessions.

Initial fine-tuning (10, 50, 200 epochs).
Our initial training strategy utilized a gene-based bucketing method to ensure representation from various gene types. We conducted trials at 10, 50, and 200 epochs. However this method failed to correct for underlying data imbalances, and by 200 epochs, the model appeared to exhibit severe overfitting: the model predicted the majority class of "Benign" or 0 exclusively, a similar behavior with the exact opposite result of the initial DT-GPT experiments.

Balanced Dataset Fine-Tuning (50, 200 epochs).
In attempt to resolve the collapse of the decision boundary, we curated a differently balanced dataset such that the number of samples labeled "benign" and "pathogenic" were equalized in a 1:1 ratio for both the training and testing dataset via random sampling. The DNABERT-2 model was then retrained on this new dataset for 50 and 200 epochs.

Evaluation.
Models were then evaluated by (1) loading them locally onto a Jupyter Notebook file to be able to generate predictions on the respective unseen testing dataset, (2) generated predictions, (3) mapped to a binary label, and (4) computed evaluation metrics: accuracy, precision, recall and F1-scores, as well as confusion matrices for better visual representations.

2.3. Baseline Predictors

As baselines, we started incorporating established in-silico pathogenicity predictors including FATHMM-MKL, SIFT and PolyPhen-2. Each predictors are expected to output a score and a prediction about whether the variants are benign or pathogenic. Then, compare the output result from each file to get the true label for analyzing the accuracy.

2.3.1. FATHMM-MKL

Functional Analysis Through Hidden Markov Models — Multiple Kernel Learning (FATHMM-MKL) is a computational method designed to predict whether a given genetic variant, typically a missense SNV, is benign or deleterious. It combines evolutionary conservation, functional genomic annotations, and regulatory information into a unified machine-learning model. Each feature represents a kernel in the model, and the machine learns how to weight each kernel to get the best prediction. Some examples of the features used by FATHMM-MKL are as follows:

- Evolutionary Conservation: Highly conserved residues are more likely to be functionally important, so substitutions at these positions may be damaging.
- Regulatory and Functional Genomic Features: Promoters, Enhancers, Histone modification marks, etc.

FATHMM-MKL is strong in integrating coding and noncoding annotations and incorporating regulatory genomic data, giving better performance for variants outside protein-coding regions. But FATHMM-MKL definitely has some drawbacks, which is it only provides scores for GRCh37, while most of our variants are on GRCh38. We tried to convert the data from GRCh38 to GRCh37 using BED format and liftOver tool. However, the result shows only two valid predictions among over 33,000 variants.

# Chromosome	Position	Ref. Base	Mutant Base	Non-Coding Score	Non-Coding Groups	Coding Score	Coding Groups	Warning
11	118926651	T	C	0.12974 ABC	0.04677 AEFDBCI			
11	118926447	G	A	0.11031 ABC	0.00894 AEFDBCI			

Fig. 3: FATHMM-MKL Result

2.3.2. SIFT

SIFT, or Sorting Intolerant From Tolerant, relies on evolutionary conservation. Functionally important amino acid positions tend to be conserved across evolution. Therefore, substitutions at highly conserved positions are more likely to be damaging. If a residue is unchanged across many species, SIFT assumes that nature has "selected" for that amino acid because it is essential for protein function.

How does SIFT works:

1. Searches protein databases to find homologs of the query protein across different species.
2. Multiple sequence alignment (MSA): For each amino acid position, SIFT counts how often each possible residue appears in evolution.
3. Calculate the probability of tolerance

SIFT outputs a score from 0 to 1, with lower scores indicating more damaging predictions. It's strengths are - It's fast, works well when good evolutionary alignments exist, and Supports SNVs. The two major limitations are - It's performance depends on the quality of homologous sequence alignments and it does not incorporate structural information.

Since SIFT runs on top of VEP (Variant Effect Predictor) tool, it gives predictions for each transcription. In other words, a variant in the genome can map to multiple transcriptions, and SIFT predicts a score for each of them. Therefore, we analyze the output score in three different ways:

1. **Average case**: Uses majority vote across transcripts.
2. **Best case**: Counts a prediction as correct if any transcript's prediction matches the truth.
3. **Worst case**: Counts the variant's prediction as incorrect if any transcript's prediction disagrees.

2.3.3. PolyPhen-2

PolyPhen-2, or Polymorphism Phenotyping version 2, predicts whether a missense variant is likely to be benign or damaging to protein structure and function. Unlike SIFT, which relies primarily on evolutionary conservation, PolyPhen-2 integrates sequence, structural, and functional information.

How does PolyPhen-2 works:

1. Identify The Protein and Residue: Maps the variant to a specific protein and amino acid position using RefSeq/UniProt data.
2. Collect Features: Including sequence, structural, and protein family information
3. Naive Bayes classifier: Calls a machine-learning classifier (Naive Bayes) trained on known pathogenic and benign missense variants.
4. Output Prediction

PolyPhen-2 also outputs a score form 0 to 1, but higher scores indicating more dangerous predictions, along with a prediction category (benign/damaging). PolyPhen is powerful at incorporating structural biology but only works on missense variants, no indels or noncoding variants

2.4. Evaluation and Metrics

Dataset splits were organized by chromosome to further minimize sequence leakage between training and testing sets. Evaluation metrics include the Area Under the Receiver Operating Characteristic Curve (AUROC), Area Under the Precision–Recall Curve (AUPR), and calibration error. These metrics allow comparison of the predictive discrimination and reliability of DNABERT and DT-GPT models relative to traditional feature-based predictors such as SIFT and PolyPhen-2.

3. Results

The aim of this project is to evaluate the effectiveness of large language model–based sequence classifiers, DNABERT and DT-GPT, in predicting whether single-nucleotide variants occurring in genes associated with single-gene disorders are pathogenic or benign. Their performance is compared against three widely used pathogenicity prediction tools: SIFT, PolyPhen-2, and FATHMM-MKL. Using a curated dataset of ClinVar-validated variants, we present side-by-side performance metrics for both the LLM-based models and the traditional predictors. Evaluation metrics include accuracy, precision, recall, F1-score, and confusion matrices. Through this comparison, we assess whether transformer-based LLMs provide improved or complementary capabilities for variant interpretation relative to existing tools, or whether traditional feature-based models remain more reliable.

3.1. DT-GPT

Across all fine-tuning configurations (unbalanced, debiased unbalanced, and balanced) the DT-GPT model consistently collapsed to predicting the "pathogenic" class for nearly every variant. This effect persisted even when the training dataset was fully balanced and all ClinVar classification text was removed prior to fine-tuning.

Evaluation of the one-epoch balanced DT-GPT model trained on Google Colab yielded an accuracy of approximately 0.50 on a balanced 300-variant test set. However, this reflected a trivial predictor as the pathogenic class was 1.0 while the benign class was 0.0, resulting in a confusion matrix in which every prediction belonged to a single class. Precision, macro-F1, and weighted-F1 all confirmed the failure to learn a functional decision boundary.

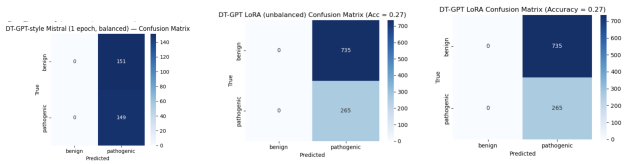

Fig. 4: DT-GPT confusion matrices across different training regimes. From left to right: (1) balanced dataset (1 epoch, Colab), (2) unbalanced dataset with ClinVar label text removed (5 epochs), and (3) original unbalanced dataset (3 epochs). In all configurations, the model collapsed toward predicting the pathogenic class almost exclusively.

3.2. DNABERT-2

gene-Stratified data Fine-Tuning Results

The result metrics of the DNABERT-2 model trained on the gene-stratified bucket dataset similar to the DT-GPT results, collapsed to predicting purely "benign" or 0 for every variant. Shockingly, the best performing model of this type was the one trained with a lower number of epochs, suggesting increasing the epochs correlates to overfitting of the data.

	10 epochs	50 epochs	200 epochs
Accuracy	0.730863	0.720535	0.720535
Precision	0.522339	0.0	0.0
Recall	0.432065	0.0	0.0
F1 Score	0.472933	0.0	0.0
AUC-ROC	0.697301	0.501676	0.474546

Table 1. Result table of DNABERT-2 Evaluation Metrics trained on gene-stratified dataset

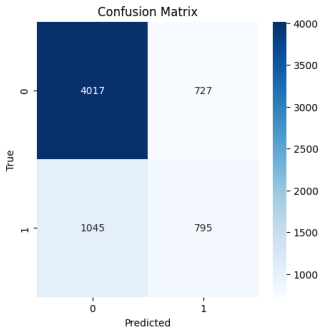

Fig. 5: Confusion Matrix of DNABERT-2 fine-tuned model on gene-stratified dataset with 10 epochs

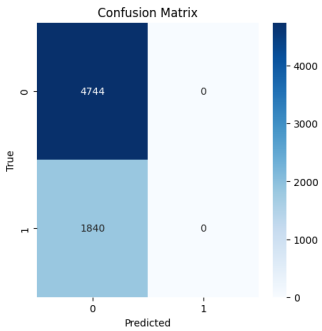

Fig. 6: Confusion Matrix of DNABERT-2 fine-tuned model on gene-stratified dataset with 200 epochs

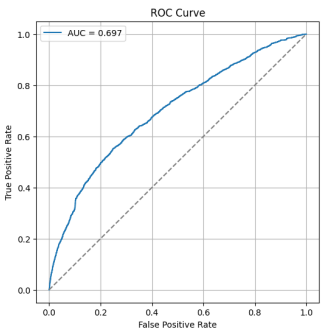

Fig. 7: ROC Curve of DNABERT-2 fine-tuned model on gene-stratified dataset with 10 epochs

strictly Balanced data Fine-Tuning Results

The evaluation metrics for the DNABERT-2 model, fine-tuned on a strictly balanced dataset (equal representation of benign and pathogenic samples in both training and test sets), aligned more closely with theoretical expectations than the previous gene-stratified models. Surprisingly, the model trained for 50 epochs exhibited metrics hovering around 0.5, indicating performance equivalent to random chance.

However, increasing the training duration to 200 epochs yielded an improvement, with metrics rising to approximately 0.6 across all categories. Notably, this performance is nearly equivalent to the results observed at 10 epochs. This non-linear progression,

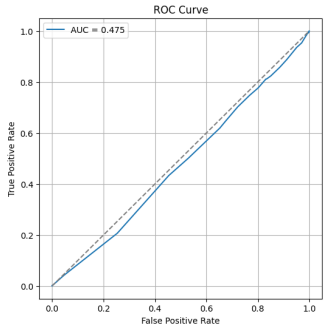

Fig. 8: ROC Curve of DNABERT-2 fine-tuned model on gene-stratified dataset with 200 epochs

where the performance has a dip in 50 epochs before recovering at 200, suggests the model may be sensitive to training duration and there exists some optimal stopping point between the early and late training stages.

	10 epochs	50 epochs	200 epochs
Accuracy	0.621274	0.493857	0.614712
Precision	0.619203	0.491153	0.619433
Recall	0.622942	0.441822	0.587816
F1 Score	0.621067	0.465183	0.603210
AUC-ROC	0.667238	0.497089	0.663226

Table 2. Result table of DNABERT-2 Evaluation Metrics trained on strictly balanced dataset

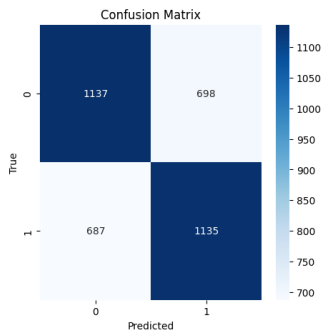

Fig. 9: Confusion Matrix of DNABERT-2 fine-tuned model on strictly balanced dataset with 10 epochs

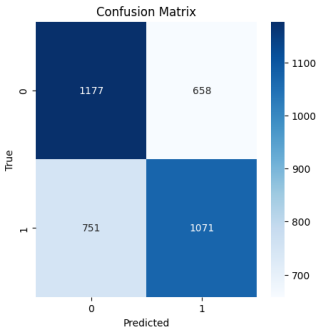

Fig. 10: Confusion Matrix of DNABERT-2 fine-tuned model on strictly balanced dataset with 200 epochs

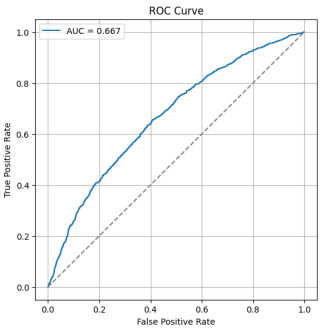

Fig. 11: ROC Curve of DNABERT-2 fine-tuned model on strictly balanced dataset with 10 epochs

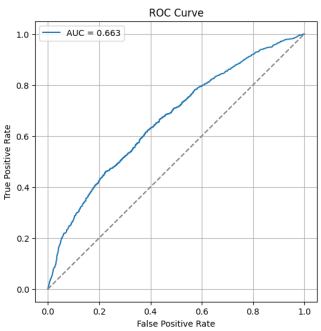

Fig. 12: ROC Curve of DNABERT-2 fine-tuned model on strictly balanced dataset with 200 epochs

3.3. SIFT

SIFT was not trained on our dataset; instead, it was used in its standard form via the Variant Effect Predictor (VEP) to score missense variants. For each variant in our dataset, SIFT returns scores per transcript, which we aggregated into three evaluation modes (average, best, and worst case). We applied SIFT both to the original test set (test.csv) and to the set used for LLM evaluation (test_balanced.csv), allowing a direct comparison between a fixed evolutionary-conservation–based tool and our fine-tuned LLMs.

Figure 13 is the average case run with the full data (test.csv), will append best case and worst case at the end of the report. Generally, the average accuracy achieves nearly 0.9.

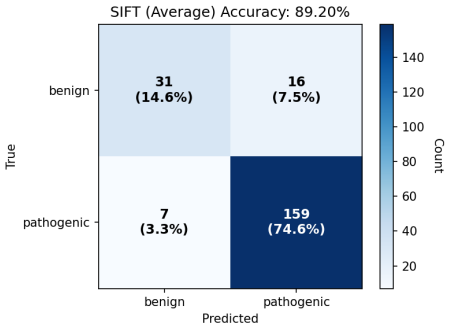

Fig. 13: SIFT Average Case Result Confusion Matrix (test.csv)

We also group the variants by Gene and observe the accuracy by the different numbers of variants in the Gene.

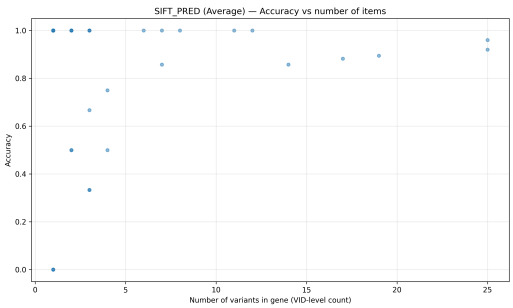

Fig. 14: SIFT Average Case Result Accuracy by Gene (test.csv)

Since the project's goal is to compare LLMs with baseline predictors, we use the same data (test data for LLMs) for running SIFT as well.

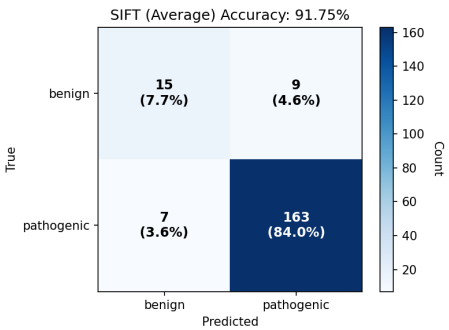

Fig. 15: SIFT Average Case Result Confusion Matrix (test_balanced.csv)

3.4. PolyPhen

Like SIFT, PolyPhen-2 was not trained or fine-tuned by us. We used the PolyPhen-2 annotations generated by VEP to obtain predictions for missense variants, and then evaluated performance on both the original set (test.csv) and the same set used for DNABERT-2 and DT-GPT (test_balanced.csv). By treating PolyPhen-2 as a fixed scoring function and comparing its confusion matrices and per-gene accuracies to those of our LLMs, we can quantify how much (or how little) the sequence-only transformers are able to match a structure- and evolution-aware baseline.

Since PolyPhen-2 also runs on top of VEP, we use the same approach to analyze the accuracy and attach the best case and worst case in the end of the report. Overall, PolyPhen-2 performs well in predicting the impact of the missense variants, achieving 0.85 accuracy in the average case.

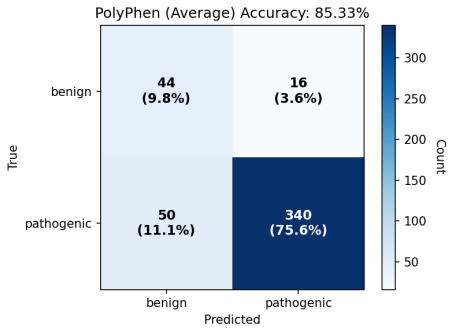

Fig. 16: PolyPhen-2 Average Case Result Confusion Matrix (test.csv)

Also, group the variants by Gene and, same as SIFT, run PolyPhen-2 with the same data testing for LLMs.

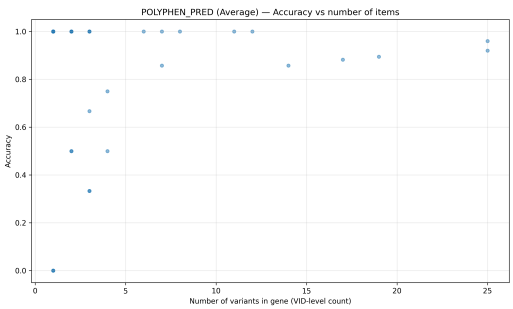

Fig. 17: PolyPhen-2 Average Case Result Accuracy by Gene (test.csv)

4. Discussion

Our results show that, in this setting, large genomic language models did not outperform traditional in-silico predictors. Both DT-GPT and DNABERT-2 were sensitive to class imbalance and

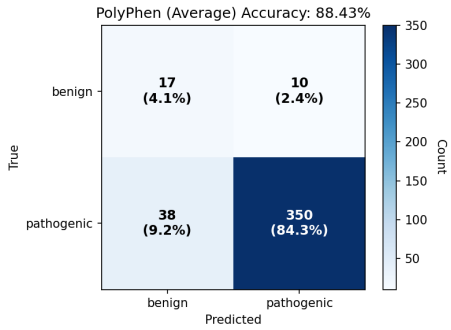

Fig. 18: PolyPhen-2 Average Case Result Confusion Matrix (test_balanced.csv)

training configuration, and frequently collapsed to trivial single-class predictions despite fine-tuning effort. By contrast, fixed baseline tools such as SIFT, PolyPhen-2, and FATHMM-MKL, which encode decades of evolutionary and structural knowledge, achieved more stable and interpretable performance on the same variants.

These findings challenge the intuitive expectation that "LLMs will always do better" simply because they are larger or more recent. Instead, they suggest that when the training signal is limited, noisy, or weakly aligned with the pretraining objective, highly parameterized sequence models can be brittle and prone to degenerate solutions. In this context, traditional feature-based models remain competitive and, in our experiments, generally superior for pathogenicity prediction on ClinVar variants.

4.1. DT-GPT Discussion

The DT-GPT component of the project yielded unexpectedly poor performance. Across all experiments, including multiple-epoch unbalanced fine-tuning, debiased fine-tuning without ClinVar classification text, and balanced-dataset fine-tuning, the model converged to predicting the "pathogenic" class for nearly every input sequence. This was surprising, as we initially expected modern genomic LLMs to outperform traditional tools given recent successes of transformer models in biological language modeling.

Several factors likely contributed to this failure mode. First, DT-GPT inherits strong priors from large-scale biomedical text pretraining, which frequently emphasizes risk, disease, and pathogenicity; these priors may overpower the relatively small supervised dataset used for fine-tuning. Second, free-form generative predictions require post-hoc string matching to extract a binary label, and even balanced training does not eliminate linguistic asymmetries in how the model discusses benign versus pathogenic variants. Third and most importantly one epoch of balanced fine-tuning is often insufficient to reorient such a large model's decision boundary.

These results underscore an important limitation of naïvely using generative LLMs as classifiers without explicit discriminative objectives or constrained output spaces, models may converge to degenerate solutions, even when trained on balanced and well-curated data. Future improvements should explore (a) training over multiple (significantly more) epochs, (b) reinforcement learning or contrastive objectives that enforce class separation,

(c) calibration techniques to mitigate safety-oriented pathogenicity bias inherited from pretraining.

4.2. DNABERT-2 Discussion

While the fine-tuned DNABERT-2 models demonstrated a greater capacity for learning than the DT-GPT models, the performances revealed significant sensitivity to class balance and training duration. In the gene-stratified experiments, the model suffered from a decision boundary collapse similar to that of DT-GPT, though with opposite polarity: it converged to predicting samples be of the "Benign" class exclusively. This indicates that even with complex architectures like DNABERT-2, slight class imbalances can drive the optimization process toward trivial local minima where the model ignores the input sequence entirely in favor of statistical priors.

The implementation of a strictly balanced dataset with a 1:1 ratio of benign to pathogenic forced the DNABERT-2 models to learn distinctive features, achieving an accuracy of ≈ 0.62 and an AUC-ROC of 0.66. Generally, the progression dynamic displayed a non-linear U-shaped progression, as the model performed relatively well at early stages (10 epochs), collapsed to a near equivalent of a random-chance performance at 50 epochs, and seemingly recovered by 200 epochs. The suggests training instability, which could be due to the hardware constraints of the local RTX 3080, which requires smaller batch sizes due to limited VRAM.

Ultimately, while the model demonstrated the capacity to distinguish pathogenic variants better than random chance, there are several shortcomings: mainly the performance ceiling ≈ 0.62 and non-linear training progression. The current results imply that a 101-bp context window might be insufficient for capturing pathogenicity signals or that a more rigorous hyperparameter tuning, which a stricter learning rate schedule, could stabilize the model during intermediate epochs.

4.3. Baseline Predictor Discussion

One thing caught our attention is that although the input data is already balanced, the output from both tools is biased toward pathogenic predictions. There are a few possible reasons:

1. VEP produces multiple rows per variant (one per transcript); a pathogenic variant tends to create more transcripts. Nevertheless, this can be avoided by averaging the predictions across transcripts, which we've already done that.

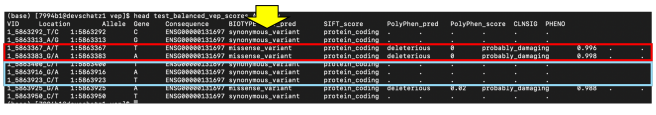

Fig. 19: All scores from VEP

2. Benign variants are often not missense, they were dropped when converting the raw score to the TSV file. We restricted our analysis to missense variants in protein-coding regions because VEP provides prediction scores for these variants that can be directly compared with the ground-truth labels in the input dataset. Other variant types do not receive SIFT or PolyPhen-2 predictions and were therefore excluded from the analysis. As seen in the figure above, synonymous variants do not get a prediction score from VEP.

- Missense variants (very high pathogenic rate): Change one amino acid to another, can affect protein structure/function.
- Synonymous variants (likely benign): Do NOT change the amino acid, no effect on protein sequence.
- Noncoding variants (usually benign): Do NOT affect protein sequence at all.
- Loss-of-function variants (Mixed): Do NOT change one amino acid, they disrupt the entire protein, but are usually not scored by these tools. (Example: SIFT uses evolutionary conservation of amino acids at the same position. For frameshift, the position doesn't exist anymore. As for nonsense, such as creating a STOP, is not a substitution. So SIFT does not apply.)

Therefore, the accuracy reported may be somewhat exaggerated. However, we can conclude that these tools perform well in predicting missense variants.

In all of our experiments, established baseline predictors such as SIFT, PolyPhen-2, and FATHMM-MKL consistently outperformed the LLM-based models. These tools perform well because they have the rich biological information that includes evolutionary conservation, protein structural features, and functional genomic annotations, which directly reflect the biochemical constraints underlying variant pathogenicity. The superior performance of the baselines highlights the importance of specific biological data, suggesting that future improvements should focus on a hybrid approaches that combine LLM representations with traditional evolutionary and structural features.

5. Acknowledgment

Professor: Michael Schatz
Teaching Assistance: Mahler Revsine

References

1. Yanrong Ji, Zhangyang Zhou, Han Liu, and Ramana V Davuluri. DNABERT: pre-trained bidirectional encoder representations from transformers model for DNA-language in genome. *Bioinformatics*, 37(15):2112–2120, 2021.
2. Michael P Menden, et al. DT-GPT: A domain-tuned generative pretrained transformer for clinical and genomic applications. *bioRxiv preprint*, 2024.
3. Alistair E. W. Johnson, Tom J. Pollard, Lu Shen, Li-wei H. Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G. Mark. MIMIC-III, a freely accessible critical care database. *Scientific Data*, 3:160035, 2016.
4. Philipp Rentzsch, Daniela Witten, Gregory M Cooper, Jay Shendure, and Martin Kircher. CADD: predicting the deleteriousness of variants throughout the human genome. *Nucleic Acids Research*, 47(D1):D886–D894, 2019.
5. N. M. Ioannidis, et al. REVEL: An ensemble method for predicting the pathogenicity of rare missense variants. *American Journal of Human Genetics*, 99(4):877–885, 2016.
6. Pauline C Ng and Steven Henikoff. SIFT: predicting amino acid changes that affect protein function. *Nucleic Acids Research*, 31(13):3812–3814, 2003.

7. Hashem A Shihab, et al. Predicting the functional, molecular, and phenotypic consequences of amino acid substitutions using hidden Markov models. *Human Mutation*, 36(1):57–65, 2015.

Supplemental Figures

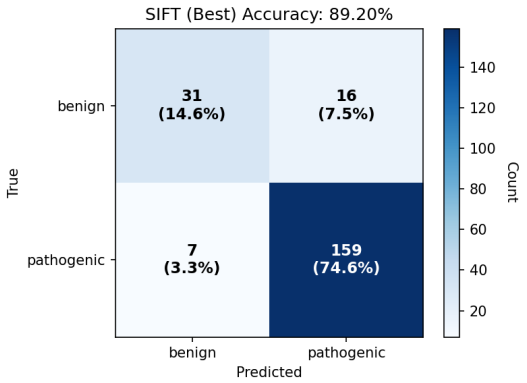

Fig. 20: SIFT Best Case Result Confusion Matrix (test.csv)

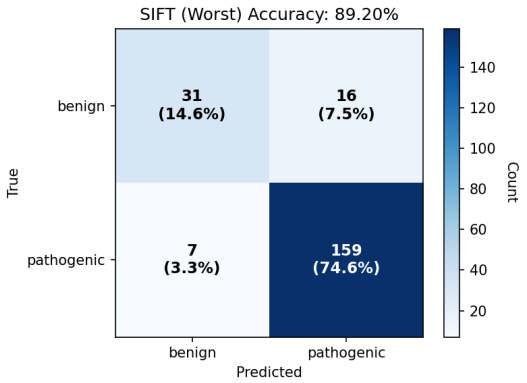

Fig. 21: SIFT Worst Case Result Confusion Matrix (test.csv)

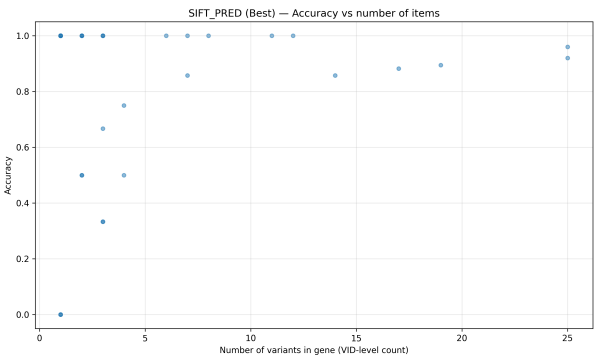

Fig. 22: SIFT Best Case Result Accuracy by Gene (test.csv)

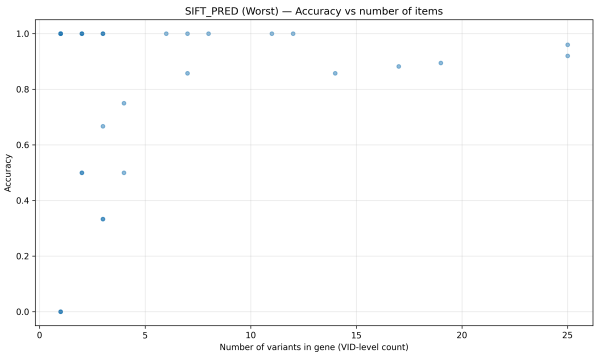

Fig. 23: SIFT Worst Case Result Accuracy by Gene (test.csv)

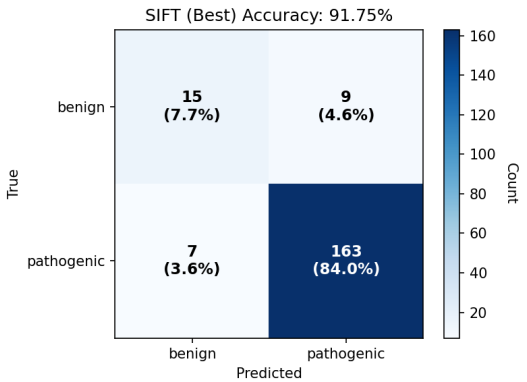

Fig. 24: SIFT Best Case Result Confusion Matrix (test_balanced.csv)

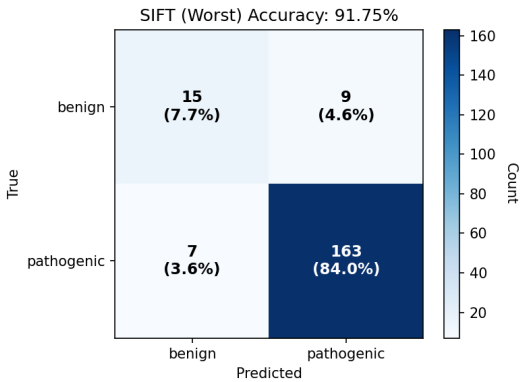

Fig. 25: SIFT Worst Case Result Confusion Matrix (test_balanced.csv)

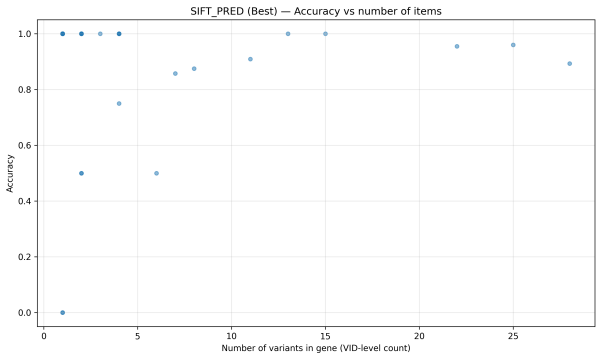

Fig. 26: SIFT Best Case Result Accuracy by Gene (test_balanced.csv)

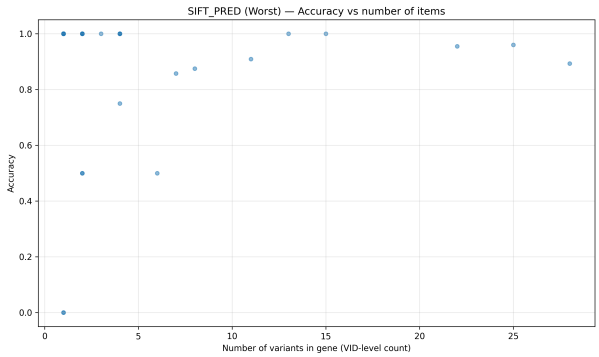

Fig. 27: SIFT Worst Case Result Accuracy by Gene (test_balanced.csv)

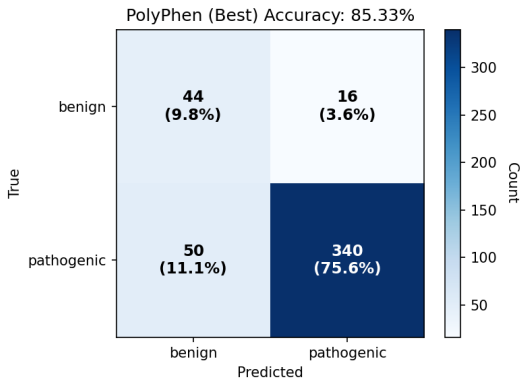

Fig. 28: PolyPhen-2 Best Case Result Confusion Matrix (test.csv)

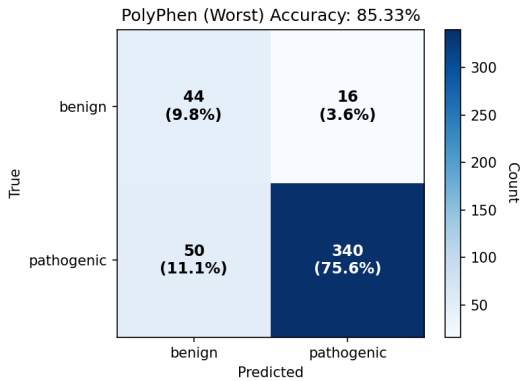

Fig. 29: PolyPhen-2 Worst Case Result Confusion Matrix (test.csv)

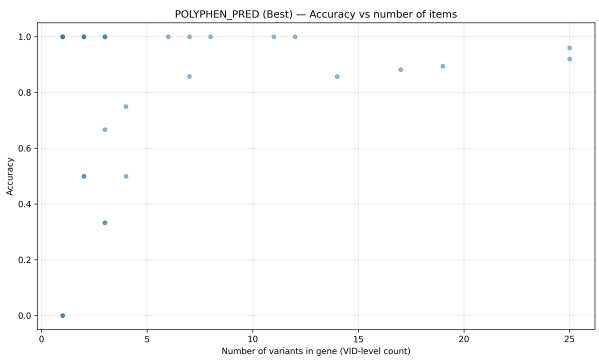

Fig. 30: PolyPhen-2 Best Case Result Accuracy by Gene (test.csv)

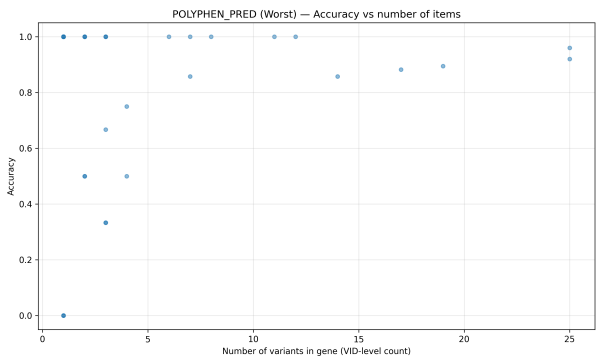

Fig. 31: PolyPhen-2 Worst Case Result Accuracy by Gene (test.csv)

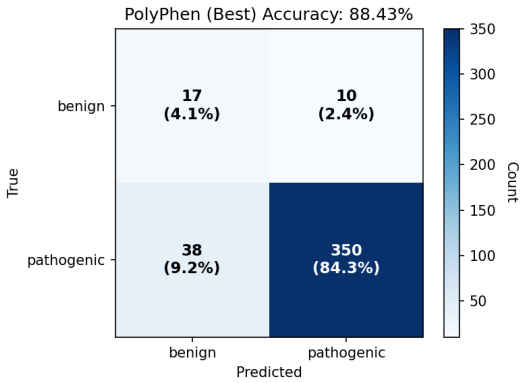

Fig. 32: PolyPhen-2 Best Case Result Confusion Matrix (test_balanced.csv)

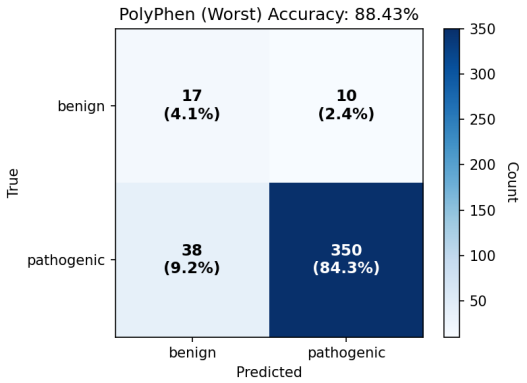

Fig. 33: PolyPhen-2 Worst Case Result Confusion Matrix (test_balanced.csv)

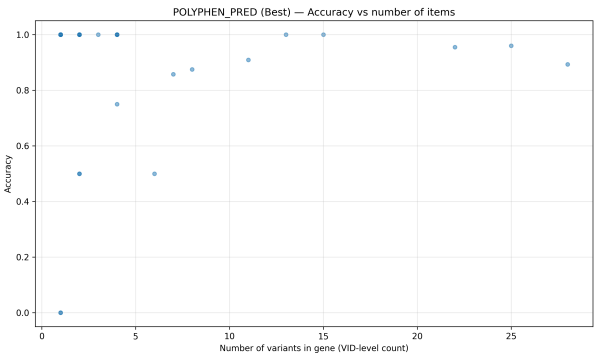

Fig. 34: PolyPhen-2 Best Case Result Accuracy by Gene (test_balanced.csv)

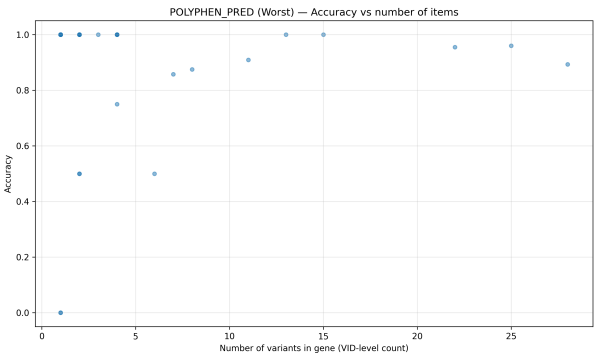

Fig. 35: PolyPhen-2 Worst Case Result Accuracy by Gene (test_balanced.csv)