

Evidence of Policy Violation by Reviewer wNfJ

Paper: #6307

1. Summary

This report presents evidence that the review provided by Reviewer wNfJ (Score: 2) was **fully generated using an LLM**, violating ICLR's review policy. Our analysis includes a control group: the same detection tools correctly identified all other reviewers (WqcR, rGrB, CKa4, F3q3) as "Fully Human," eliminating the possibility that the paper's technical jargon triggered a false positive. The summary of detection results is shown in Table 1.

Table 1: Summary of LLM detection result.

Reviewer	Rating	Pangram Labs	GPTZero
wNfJ	2	Fully AI-generated	100% AI generated
WqcR	4		
rGrB	6		
CKa4	6		100% Human
F3q3	6		

2. Evidence

2a. Pangram Labs

Pangram Labs (<https://www.pangram.com/>) is a leading enterprise-grade detection solution trusted by global organizations and academic institutions. Notably, Pangram provides a specialized detection module for ICLR reviews (<https://iclr.pangram.com>), specifically calibrated to handle the technical density and formal tone of ICLR submissions.

As shown in Figure 1, the tool successfully filtered our submission's reviews. Only **Reviewer wNfJ** (Score: 2) was flagged as "**Fully AI-generated**" (Red). The table reveals a perfect correlation between the AI flag and the low quality: the AI-generated review corresponds to the outlier score of 2.00, whereas the verified human reviews maintain a high average rating of 5.50.

Summary Statistics

EditLens Prediction	Count	Avg Rating	Avg Confidence	Avg Length (chars)
Fully AI-generated	1 (20%)	2.00	4.00	4129
Heavily AI-edited	0 (0%)	N/A	N/A	N/A
Moderately AI-edited	0 (0%)	N/A	N/A	N/A
Lightly AI-edited	0 (0%)	N/A	N/A	N/A
Fully human-written	4 (80%)	5.50	4.00	2367
Total	5 (100%)	4.80	4.00	2719

Figure 1: Screen shot from Pangram Labs detection result for paper # 6307.

Official report link: https://iclr.pangram.com/reviews?submission_number=6307

2b. GPTZero

GPTZero (<https://gptzero.me/>) is widely regarded as the "gold standard" for AI detection, used by educators and institutions worldwide. It utilizes perplexity and burstiness metrics to distinguish between human and machine writing patterns.

Figure 2 identifies **Reviewer wNfJ** as "**100% AI-generated.**" In contrast, Figures 3 and 4 confirm that reviews from other reviewers were correctly identified as "**100% Human.**"

This paper introduces TD-HNODE, a disease progression model that integrates clinical knowledge into a temporally detailed hypergraph combined with Neural ODE. Each node represents a disease complication marker, and each hyperedge is a predefined clinically validated progression trajectory. For disease modeling, the authors propose a time-adaptive Laplacian that governs continuous-time diffusion of latent marker states, comprising an attention-based incidence matrix for patient-specific, time-aware weighting of markers and a learnable hyperedge weight matrix. Experiments on two EHR datasets (University Hospital and MIMIC-IV) show that TD-HNODE improves accuracy, recall, and F1 compared with strong baselines (T-LSTM, Contiformer, TGNE, HyperTime, CODE-RNN). Ablations support the contributions of both adaptive incidence and learnable weights. Continuous-time modeling of chronic disease trajectories is an important and emerging topic. Interpretability might be good, as the hyperedges align with known clinical pathways. This might serve as a foundation for explanation and clinical validation. Consistent improvements on two EHR datasets, particularly in recall (clinically critical for early detection). Over-complex and arguably unnatural construction. While the idea of embedding medical knowledge into continuous-time dynamics is valuable, the resulting architecture feels heavily engineered. TD-HNODE stacks many modeling layers—curated trajectories dense inter-trajectory weighting ODE integration—each adding parameters without clear generative justification. It is difficult to discern whether the model captures meaningful structure or merely benefits from large capacity. Everything is learned, risking loss of inductive bias. Almost all structural components time encodings are trainable. This undermines the "knowledge-infused" motivation: the learned Laplacian may diverge from curated pathways, reducing interpretability. Regularization toward clinical priors or partial parameter freezing would help maintain domain grounding. Bias compounding across multiple submodules. The architecture effectively stacks a model on top of another (attention self-attention pooling decoder). Each stage may introduce its own bias, and the composition could amplify rather than mitigate error. It remains unclear which layer drives performance versus redundancy. Questionable scalability of the hypergraph construction. The temporally detailed hypergraph may require combining all observed markers across trajectories, potentially leading to a combinatorial explosion of hyperedges. The paper does not quantify computational or memory costs beyond brief remarks in the appendix. Recomputing dense, time-varying could be infeasible for large EHRs.

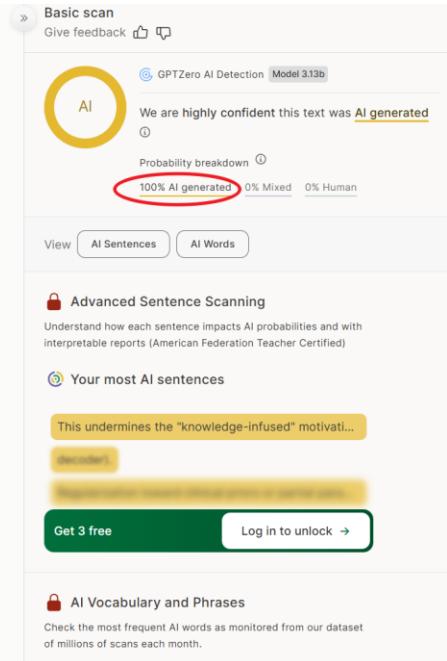


Figure 2: GPTZero detection result (100% AI generated) for **Reviewer wNfJ**.

This paper addresses the prediction of disease progression from patient encounter data, incorporating risk factors such as medications, laboratory test results, and vital signs. The authors propose TD-HNODE, which combines a neural ODE and a hypergraph neural network, where Learnable TD-Hypergraph Laplacian plays a key role to make the model more data-driven and adaptable to patient-specific disease trajectories while maintaining its clinically verified nature. The proposed method was evaluated on two real-world EHR datasets and consistently outperformed baselines. -- Learnable TD-hypergraph Laplacian is a reasonable enhancement for the combination of neural ODE and hypergraph neural network, where Attention-based Incidence Matrix adjusts the degree of attributions of v in e flexibly according to the context, and Learnable Hyperedge Weight Matrix captures data-driven similarities between trajectories. -- Experimental results on multiple datasets demonstrated the effectiveness of the proposed method. -- The case study is interesting and practically important.- Clarity issues: In I.100, what is k? In I.147, The authors mentioned "we use the terms 'hyperedge', 'pathway', and 'trajectory' interchangeably", but this makes descriptions confusing. For example, p_j and e_j look the same, so we should use only one of them consistently throughout the paper. In Eq.2, LHS is better to be f(t, S(t), x(t))/\Theta not dS(t)/dt. In Eq.3, I is not defined. In the descriptions starting from I.242, e is used like an index of edge, but it was j until then. The index for the edge should be j, and the edge itself is denoted as e for consistency. In I.265, the temporally detailed hyperedge should have u on superscript. In I.269, e may not be the index for O and F, maybe. For indexing, only j is enough.

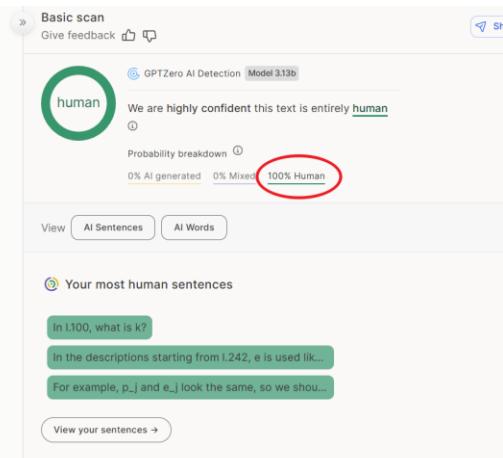


Figure 3: Control check: GPTZero detection result (100% Human) for Reviewer rGrB.

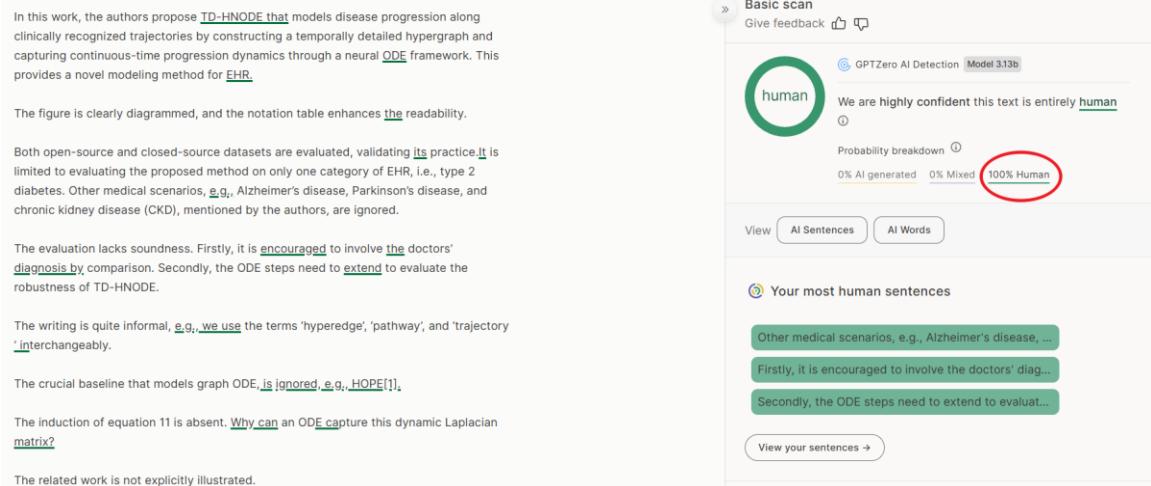


Figure 4: Control check: GPTZero detection result (100% Human) for Reviewer WqcR.

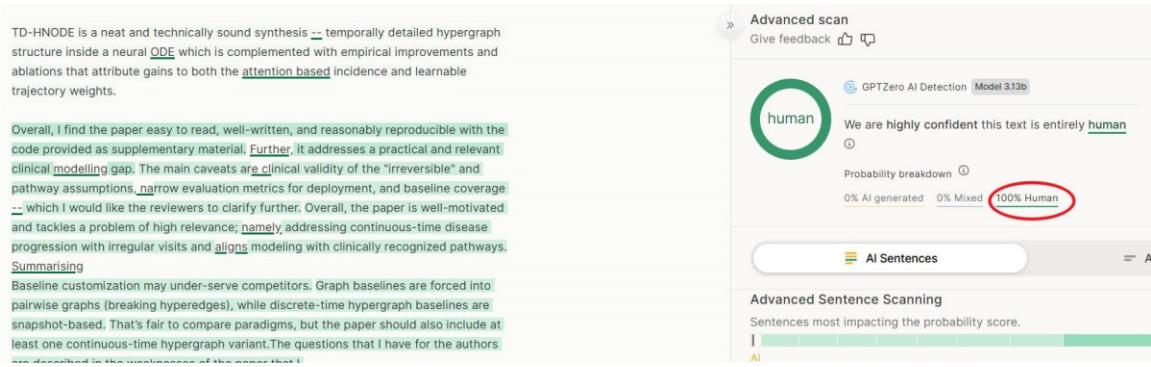


Figure 5: Control check: GPTZero detection result (100% Human) for Reviewer CKa4.

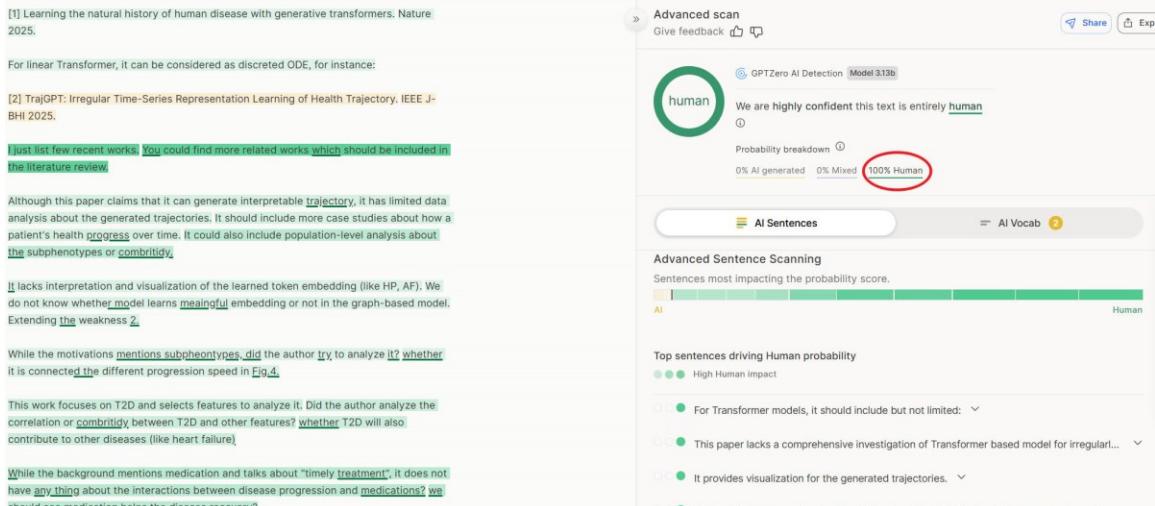


Figure 6: Control check: GPTZero detection result (100% Human) for Reviewer F3q3.