SynPro: Measuring Syntactic Process Divergence in Human and ChatGPT-Generated Text

Anonymous ACL submission

Abstract

Large language models can produce fluent text with few obvious grammatical errors, but fluent output need not reproduce the syntactic process of human writing. We introduce SynPro, a framework for comparing human and LLM-generated syntax through dependency-label paths. Rather than only counting which dependency relations appear, SynPro measures how those relations compose along root-to-leaf paths using marginal entropy, transition entropy, and Jensen–Shannon divergence over labels, transitions, and full path types. Across HC3 and M4, human and ChatGPT text have similar dependency-label marginals but diverge more strongly in dependency composition. In HC3, path-type JS divergence is $16.7\times$ larger than marginal-label JS; in M4, path-type JS remains $7.5\times$–$10.0\times$ larger under sentence-length matching, joint length-and-depth matching, and paired token-budget truncation. Human text also has consistently higher transition entropy than ChatGPT text, indicating less predictable local dependency transitions. Additional support controls show that the path-level gap persists after equalizing path counts and filtering rare path types. These results suggest that generated text can match human writing in syntactic ingredients while differing in how those ingredients are assembled.

1 Introduction

Large language models (LLMs) now produce text that is often fluent, coherent, and locally grammatical. This makes human–LLM comparison difficult. Because generated text uses plausible words and contains few obvious grammatical errors, it raises the question of how we measure syntactic differences. Our study addresses a more specific variation of this problem, examining whether human and LLM-generated texts compose those relations in the same way when using similar syntactic structures.

We focus on dependency syntax. A dependency parse represents a sentence as a tree whose labeled edges encode relations such as `nsubj`, `obj`, `amod`, and `prep`. Many syntactic comparisons aggregate these labels into marginal or sentence-level features such as dependency-label frequencies, parse depth, dependency length, readability, and formal grammar-type counts. These features describe the syntactic ingredients of a corpus, but they don't fully describe syntactic composition and how dependency relations transition along tree paths.

This matters because two corpora can have similar dependency-label marginals while differing in their dependency paths. For instance, human and generated text may use subjects, objects, and modifiers at comparable rates, yet combine them into different local transition patterns or root-to-leaf path distributions. SynPro treats dependency paths as samples from a stochastic process over relation labels and compares human and LLM-generated text using marginal dependency entropy, first-order transition entropy, and Jensen–Shannon (JS) divergence over label, transition, and path-type distributions.

Our claim is not that LLM syntax is uniformly simpler, nor that every entropy-based diagnostic moves in the same direction. We have the narrower conclusion that across HC3 and M4, human and ChatGPT text have relatively similar dependency-label marginals, but differ in local dependency-transition entropy and full dependency-path distributions. In M4, the transition-entropy and path-divergence effects persist under sentence-length matching, joint length-and-depth matching, and paired token-budget truncation. On the other hand, syntactic predictability gain and rare-path mass are more corpus-sensitive, so we treat them as secondary diagnostics. This paper makes three contributions:

1. We introduce SynPro, a process-level compar-

ison of human and LLM-generated syntax using root-to-leaf dependency-label paths.

2. We distinguish dependency marginals from dependency composition using transition entropy and JS divergence over labels, transitions, and paths.

3. We show on HC3 and M4 that human and ChatGPT text can be close in marginal dependency labels while diverging in transition entropy and dependency-path structure, even under length and depth controls.

Section 2 situates SynPro against syntactic analyses of generated text, detection work, and information-theoretic syntax. Sections 3–4 define the method and experiments, Section 5 presents results, and Section 6 discusses implications.

2 Related Work

Syntactic analyses of generated text. Ju et al. (2025) study domain regeneration by prompting an open-source LLM to regenerate Wikipedia and news text and then comparing the generated and original domains using sentence length, readability, dependency-tag distributions, parse depth, and parse complexity. They find that regenerated distributions often have shifted means, lower variance, and reduced long tails relative to the human originals. Zamaraeva et al. (2025) compare New York Times-style text generated by six LLMs against real human-authored NYT writing using Head-driven Phrase Structure Grammar, showing systematic differences in the distributions of HPSG grammar types. Broader linguistic analyses also compare human and generated text through lexical, grammatical, stylistic, and domain-level properties (Muñoz-Ortiz et al., 2024; Guo et al., 2024). These studies show that syntax contains signal about text origin and domain regeneration, but their measurements are primarily marginal, sentence-level, grammar-type, or feature-distribution based. We instead compare the process by which dependency relations compose along tree paths.

AI-generated text detection. Generated-text detection has been studied through likelihood and perturbation methods, watermarking, benchmark construction, and robustness analyses (Gehrmann et al., 2019; Zellers et al., 2019; Ippolito et al., 2020; Mitchell et al., 2023; Kirchenbauer et al., 2023; Sadasivan et al., 2023; Li et al., 2024; Wu et al., 2025). Ahmed and Hammond (2026) show that dependency labels alone can be useful for detecting AI-generated text. We are not trying to optimize a detector, but instead ask whether human and generated text differ in dependency composition after marginal syntactic inventories and length effects are considered.

Information-theoretic syntax. Entropy and related quantities have long been used to study language. Genzel and Charniak (2002) propose entropy-rate constancy in text, while Verma et al. (2023) revisit such claims with neural language models. In dependency and corpus syntax, Futrell et al. (2015b) use entropy-like measures to study word-order freedom, while De Gregorio et al. (2024) use information-theoretic syntactic distances over POS sequences to study language relatedness. Moscoso del Prado Martín (2025) develops derivational entropy rates for measuring grammatical diversity. We use standard information-theoretic quantities (Shannon and Weaver, 1949; Cover and Thomas, 2006), and the contribution is their application to dependency-label paths for human–LLM syntactic comparison.

Dependency syntax and syntactic complexity. Dependency parsing provides compact head-dependent representations of sentence structure (Nivre, 2008; de Marneffe et al., 2006; Chen and Manning, 2014; Kiperwasser and Goldberg, 2016). Universal Dependencies and related frameworks enable large-scale cross-corpus comparisons (Nivre et al., 2020; de Marneffe et al., 2021). Quantitative dependency-syntax work has studied dependency length, locality, word order, parse depth, and structural complexity (Liu, 2008; Futrell et al., 2015a, 2020; Jing et al., 2021). Our work complements this by studying uncertainty and divergence in sequences of dependency relations along tree paths.

3 Syntactic Process Entropy

3.1 Dependency paths

Let a sentence be represented by a labeled dependency tree

$$T = (V, E, \ell), \tag{1}$$

where V is the set of tokens, $E \subseteq V \times V$ is the set of directed head-dependent edges, and $\ell : E \to \mathcal{R}$ maps each edge to a dependency relation label in a finite set $\mathcal{R}$. We extract root-to-leaf paths from

Unit	Example
Label	nsubj, obj, amod
Transition	obj→amod
Path type	ROOT→obj→amod

Table 1: Units of analysis in SYNPRO. Marginal statistics operate on labels; process statistics operate on transitions and path types.

each tree. Each path gives a sequence

$$r_1, r_2, \ldots, r_m, \qquad r_i \in \mathcal{R}. \tag{2}$$

We use only dependency labels and tree structure; token identities are not used. ROOT is included during path extraction for consistency but excluded from entropy and divergence estimates. Punctuation dependencies are retained, since punctuation can participate in the parser's dependency structure and may affect path distributions.

Root-to-leaf paths are a deliberately simple representation of syntactic process as they do not encode lexical choice, word order directly, or semantic roles beyond the dependency labels themselves. This restriction is useful for our setting because it makes the comparison conservative, meaning that if human and ChatGPT text differ even after tokens are discarded and only dependency-label paths remain, the difference is not reducible to surface vocabulary. At the same time, root-to-leaf paths preserve more structure than isolated dependency-label counts as they retain how local relations compose into larger dependency templates, which is the distinction between syntactic ingredients and syntactic composition that SYNPRO is designed to measure.

3.2 Entropy and divergence metrics

Let R be a dependency label sampled from all non-ROOT labels in extracted paths. The marginal dependency entropy is

$$H_1 = -\sum_{r \in \mathcal{R}} p(r) \log p(r). \tag{3}$$

This measures the diversity of dependency labels, but ignores path context.

To measure local composition, let (R_t, R_{t+1}) be adjacent labels along a path. The first-order transition entropy is

$$\begin{aligned} H_2 &= H(R_{t+1} \mid R_t) \\ &= -\sum_{a \in \mathcal{R}} p(a) \sum_{b \in \mathcal{R}} P(b \mid a) \log P(b \mid a). \end{aligned} \tag{4}$$

Lower H_2 means that the next dependency relation is more predictable from the current relation. Unlike H_1, which measures syntactic ingredients, H_2 measures local dependency composition.

We also compare full distributions using JS divergence. For distributions P and Q,

$$\begin{aligned} \mathrm{JS}(P, Q) &= \tfrac{1}{2}\,\mathrm{KL}(P \| M) \\ &\quad + \tfrac{1}{2}\,\mathrm{KL}(Q \| M), \\ M &= \tfrac{1}{2}(P + Q). \end{aligned} \tag{5}$$

We compute JS divergence between human and ChatGPT text over dependency-label marginals, adjacent transitions (R_t, R_{t+1}), and full root-to-leaf path types and report it in bits. The key comparison is label JS versus path-type JS because if label JS is small but path-type JS is much larger, the corpora use similar dependency ingredients but arrange them differently.

The empirical distributions are estimated directly from extracted paths. For a corpus C, let $N_C(r)$ be the number of non-ROOT occurrences of label r, let $N_C(a, b)$ be the number of adjacent label transitions $a \to b$, and let $N_C(\rho)$ be the number of complete root-to-leaf path types ρ. We use

$$\begin{aligned} \hat{p}_C(r) &= \frac{N_C(r)}{\sum_{r'} N_C(r')}, \\ \hat{p}_C(a, b) &= \frac{N_C(a, b)}{\sum_{a', b'} N_C(a', b')}, \\ \hat{q}_C(\rho) &= \frac{N_C(\rho)}{\sum_{\rho'} N_C(\rho')}. \end{aligned} \tag{6}$$

We use these empirical distributions for label, transition, and path-type divergences. Thus we have that $\hat{p}_C(r)$ ignores all path context, $\hat{p}_C(a, b)$ keeps local composition, and $\hat{q}_C(\rho)$ keeps complete path templates.

3.3 A sanity-check example

The difference between marginals and composition is easiest to see in a toy case. Suppose one corpus places most mass on paths of the form $a \to b \to c$, while another places most mass on $a \to c \to b$. The two corpora can have the same marginal counts of a, b, and c, and therefore the same H_1, while having different transition distributions and different path-type distributions. In this case, a marginal dependency-label comparison would miss the difference, but H_2 and path-type JS would capture it

Algorithm 1 SYNPRO: dependency-path process comparison

Require: Human corpus $\mathcal{D}_H$, LLM corpus $\mathcal{D}_L$, parser π
for $\mathcal{D} \in \{\mathcal{D}_H, \mathcal{D}_L\}$ **do**
  Parse sentences with π to obtain dependency trees.
  Extract root-to-leaf dependency-label paths.
  Remove ROOT labels from metric computations.
  Estimate label marginals $p(r)$.
  Estimate transition probabilities $P(b \mid a)$.
  Compute H_1, H_2, and path-type distribution $q(p)$.
end for
Compute JS divergence over labels, transitions, and paths.
Run marginal-preserving null models and length/depth controls.
return entropy gaps, JS divergences, and null statistics.

3.4 Secondary diagnostics

We also compute higher-order conditional entropy and syntactic predictability gain. For order k,

$$\begin{aligned} \mathrm{SPG}_k = H(R_{t+1}) \\ - H(R_{t+1} \mid R_t, \ldots, R_{t-k+1}). \end{aligned} \tag{7}$$

This is a mutual-information-style measure of how much local syntactic context reduces uncertainty about the next relation. We also compute rare-path mass, the probability mass assigned to dependency-path types below a frequency threshold. These diagnostics are useful but less stable in our experiments than H_2 and path-type JS, so we report them as secondary analyses.

4 Experimental Setup

4.1 Datasets

We evaluate on HC3 (Guo et al., 2023) and M4 (Wang et al., 2024). For HC3, we use the first 500 paired questions from `all.jsonl`. Because the file is domain-ordered, these examples are all from the `reddit_eli5` subset. For each question, we use the first human answer and the first ChatGPT answer, giving 500 human texts and 500 ChatGPT texts.

For M4, we use the `wikipedia_chatgpt` subset which consists of 1,000 paired human and ChatGPT Wikipedia-style articles with built-in source IDs. M4 differs from HC3 in both genre and length profile as it is expository Wikipedia-style text, and the human articles are substantially longer than the ChatGPT generations. Table 2 summarizes the parsed corpora.

We use the HC3 and M4 datasets in accordance with their public dataset terms and cite their creators. HC3 is distributed under CC-BY-SA except where source datasets impose stricter terms, and spaCy is released under the MIT license.

Corpus	Pairs	Human sent.	GPT sent.	Labels
HC3	500	3,253	4,394	44
M4	1,000	33,450	14,428	44

Table 2: Datasets after sentence segmentation and dependency parsing. HC3 uses paired `reddit_eli5` questions; M4 uses paired Wikipedia–ChatGPT articles.

4.2 Parsing and preprocessing

All texts are parsed with spaCy (Honnibal et al., 2020), using `en_core_web_sm`. The resulting labels follow spaCy's ClearNLP-derived dependency inventory rather than a pure Universal Dependencies conversion. We remove text spans shorter than 20 characters and parsed spans with zero tokens. Punctuation dependencies are retained. ROOT is included during path extraction but excluded from entropy and divergence estimates. Entropies are estimated by plug-in maximum likelihood with no smoothing and are reported in nats, while JS divergences are reported in bits.

4.3 Nulls and controls

We use two marginal-preserving null models. The first globally shuffles dependency labels while preserving path lengths and the overall label multiset. The second shuffles labels within each path, preserving each path's label multiset and length. Each null is repeated 100 times, and we compute bootstrap confidence intervals using 200 bootstrap samples.

Because M4 human documents are much longer than ChatGPT generations, we run three additional M4 controls. First, we match sentences by token-length bins. Second, we jointly match by sentence-length and parse-depth bins. Third, for each paired source ID, we truncate the human document at sentence boundaries to approximately match the paired ChatGPT token budget.

For sentence-length matching, we use token-length bins $[1, 10]$, $[11, 20]$, $[21, 30]$, $[31, 40]$, $[41, 60]$, and $[61, +\infty)$. For length-and-depth matching, we cross these with parse-depth bins $(0, 2)$, $(3, 4)$, $(5, 6)$, $(7, 9)$, and $(10, 100)$, and within each bin or cell, we sample equal numbers of human and ChatGPT sentences without replacement and repeat the matched analysis 100 times.

Setting	ΔH_2	Label JS	Path JS
HC3	0.172	0.016	0.266
M4 raw	0.169	0.045	0.259
M4 length	0.129	0.027	0.209
M4 length+depth	0.077	0.021	0.207
M4 trunc.	0.106	0.030	0.225

Table 3: Main process-level results. ΔH_2 is human minus ChatGPT transition entropy. JS divergences are in bits.

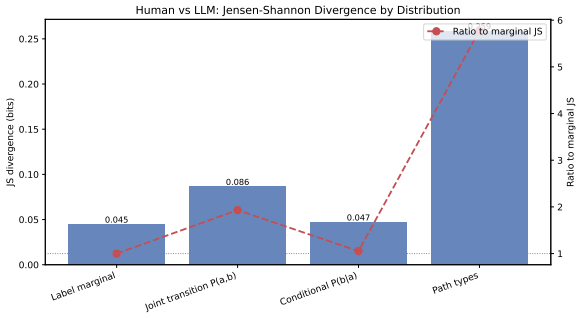

Figure 1: M4 divergence by syntactic level. Path-type distributions diverge more strongly than marginal dependency-label distributions, supporting the distinction between dependency ingredients and dependency composition.

5 Results

5.1 Marginals are close, but paths diverge

Table 3 gives the main process-level results. In HC3, the marginal dependency entropies are nearly identical: $H_1 = 3.092$ for human answers and $H_1 = 3.061$ for ChatGPT answers. The marginal label JS divergence is also small, only 0.016 bits. Thus, at the level of individual dependency labels, human and generated text use very similar syntactic inventories.

The picture changes at the path level. In HC3, full path-type JS divergence is 0.266 bits, or $16.7\times$ the marginal label divergence. M4 shows the same qualitative pattern, as raw M4 path-type JS is 0.259 bits, compared with label JS of 0.045 bits. This supports the idea that human and generated text may use similar dependency labels while arranging them into substantially different dependency-path distributions.

5.2 Transition entropy replicates across corpora

The most stable scalar process metric is first-order transition entropy, $H_2 = H(R_{t+1} \mid R_t)$. In both corpora, human text has higher transition entropy than ChatGPT text. The human–LLM gap is nearly identical across the two settings with $\Delta H_2 \approx 0.172$ in HC3 and $\Delta H_2 \approx 0.169$ in M4. This replication is noteworthy given the corpora differ in genre and structure, with paired question answering in HC3 and Wikipedia-style expository text in M4.

Lower H_2 for ChatGPT means that, given the current dependency relation, the next relation along a dependency path is more predictable. This does not imply that generated text is ungrammatical or lower quality, but simply indicates that local dependency composition differs from human text, even when marginal label inventories are similar.

5.3 Path divergence increases with syntactic structure

Figure 1 shows the M4 divergence pattern across syntactic levels. The central trend is that divergence grows as we move from individual dependency labels to richer syntactic structure. Marginal label distributions are comparatively close, while transition and path-type distributions diverge more strongly.

This pattern matches the HC3 result. Hence, marginal dependency labels describe which syntactic relations appear, while path-type distributions describe how those relations are assembled into larger structures.

5.4 Length controls preserve the M4 effect

With M4, we have the issue that human Wikipedia articles are substantially longer than ChatGPT generations. We therefore repeat the M4 analysis under three controls. First, we match sentences by token-length bins. Second, we jointly match by length and parse depth. Third, we truncate each human article at sentence boundaries to approximately match the token budget of the paired ChatGPT article.

The transition-entropy gap survives all three controls. Under sentence-length matching, $\Delta H_2 = 0.129$ with a 95% interval of $[0.123, 0.136]$. Under joint length-and-depth matching, the gap shrinks but remains positive, $\Delta H_2 = 0.077$ with interval $[0.071, 0.083]$. Under paired token-budget truncation, $\Delta H_2 = 0.106$. Therefore, length and depth explain part of the raw M4 gap, but not all of it.

The path-divergence result also survives, as across the three controls, path-type JS remains $7.5\times$–$10\times$ larger than marginal label JS. This sug-

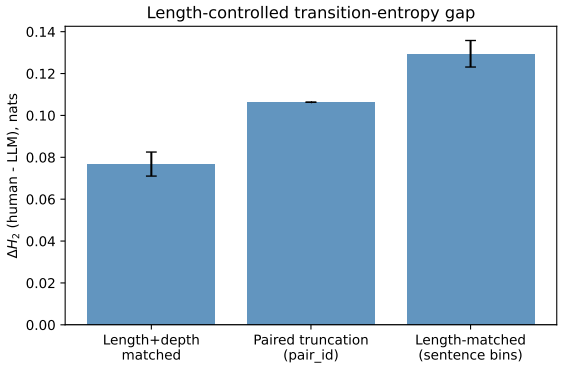

Figure 2: M4 length-controlled transition-entropy gaps. Human text has higher H_2 than ChatGPT text under sentence-length matching, joint length-and-depth matching, and paired token-budget truncation.

Transition	Human	GPT	Diff.
punct→dep	0.913	0.000	+0.913
punct→prep	0.022	0.593	−0.570
advmod→prep	0.364	0.734	−0.371
nummod→dep	0.338	0.000	+0.338
nummod→quantmod	0.136	0.423	−0.287
compound→dep	0.253	0.000	+0.253
punct→compound	0.010	0.259	−0.249
nmod→cc	0.042	0.236	−0.194

Table 4: Largest M4 conditional transition-probability differences among common dependency labels. Values are estimated from the M4 human and ChatGPT transition matrices.

gests that path-level divergence is not only a consequence of longer human documents producing more paths.

5.5 Which dependency patterns drive the effect?

The aggregate metrics show that human and ChatGPT text diverge more strongly over dependency composition than over individual dependency labels. To make this result more interpretable, we inspect transition-level differences and path-type contributions in M4. Table 4 reports the largest conditional-probability gaps $P(b \mid a)$ among common dependency labels. Table 5 reports the largest empirical contributors to path-type JS divergence.

These rows are not standalone diagnostics of authorship. A large gap for a particular transition or path type can reflect genre conventions, parser behavior, punctuation handling, or the distribution of antecedent labels. Their purpose is instead to decompose the aggregate effect: the H_2 and path-JS gaps correspond to concrete differences in continuation probabilities and recurring dependency-path templates, not only to abstract scalar summaries.

Path type	Human	GPT	JS contrib.
punct→dep	0.0242	0.0000	0.0121
punct	0.0788	0.1147	0.0024
appos→punct	0.0050	0.0005	0.0016
attr→det	0.0060	0.0154	0.0015
attr→amod	0.0041	0.0111	0.0012
appos→compound	0.0031	0.0002	0.0011
dep	0.0025	0.0001	0.0010
nsubj→compound→dep	0.0018	0.0000	0.0009

Table 5: Largest M4 path-type contributions to Jensen–Shannon divergence. Frequencies are empirical proportions within each source-specific path distribution, and contributions are measured in bits.

Several high-contribution rows involve punctuation or the generic dep label, so Tables 4 and 5 should be read as interpretive decompositions rather than independent robustness claims. Even with that caveat, they illustrate that the two corpora are not just choosing different dependency labels. They assign different continuation probabilities after a given dependency relation and place different mass on recurring dependency-path templates. The difference appears in the local and path-level process by which dependency labels are composed.

5.6 Support controls for path-type divergence

A potential concern is that path-type JS may be inflated by support-size effects. Full root-to-leaf paths are more numerous than individual dependency labels, so a corpus with more paths or a longer tail of rare paths can have larger empirical divergence even when the underlying syntactic process is similar. In our parsed M4 subset, the raw corpora have 494,619 human paths and 182,565 ChatGPT paths, with 78,482 and 23,612 unique path types, respectively. This motivates direct support controls in addition to the length and depth controls above.

Table 6 tests whether path-type JS is driven only by unequal path counts or rare path types. Equalizing the number of extracted paths does not reduce the effect: path-type JS remains 0.269 bits, about 6.0× the marginal-label JS. Filtering rare path types lowers path-type JS, as expected, but the path/marginal gap remains. With minimum pooled count thresholds of 2 and 5, path-type JS remains 4.3× and 3.6× the label JS, respectively. Even when restricting to the top 1,000 pooled path types, path-type JS remains 2.9× the label JS.

These controls show that path-level divergence is not only a consequence of the raw M4 length imbalance or one-off rare path types. The long tail contributes to the magnitude of path JS, but the

Setting	Label JS	Trans. JS	Path JS	Ratio
Raw	0.045	0.086	0.259	5.8×
Equal path count	0.045	0.087	0.269	6.0×
Min count ≥ 2	0.044	0.083	0.189	4.3×
Min count ≥ 5	0.041	0.077	0.149	3.6×
Top-1,000 paths	0.038	0.070	0.109	2.9×

Table 6: M4 finite-sample and support controls for JS divergence. Path-type divergence decreases when rare path types are filtered, but remains several times larger than marginal-label divergence.

Corpus	Metric	Global z	Within z
HC3	ΔH_2	60.1	39.0
HC3	$\Delta\,\mathrm{SPG}_1$	-238.0	-46.0
M4	ΔH_2	48.0	41.2
M4	$\Delta\,\mathrm{SPG}_1$	-13.0	-44.1

Table 7: Marginal-preserving null-model comparisons. The observed transition-entropy gap is far from the null in both corpora, indicating that the effect is not explained by label frequencies alone.

marginal-versus-process gap persists after equal-count sampling and support restriction.

5.7 Null models and secondary diagnostics

We next test whether the observed differences could be explained by marginal dependency-label frequencies alone. In HC3, the observed structure is far outside both marginal-preserving null distributions, and for the SPG_1 gap, the global-shuffle and within-path-shuffle nulls give large-magnitude z-scores ($z = -238$ and $z = -46$). M4 shows the same qualitative pattern for transition structure, as the observed transition-entropy gap remains far from marginal-preserving baselines. Table 7 reports the headline null comparisons. Here z is computed relative to the corresponding null distribution of human-minus-ChatGPT metric gaps.

We also verified that the main conclusions are not driven solely by the raw path-count imbalance in M4. Human articles yield more extracted paths and more unique path types than ChatGPT articles, but the controlled comparisons in Table 6 reduce the length and depth imbalance while preserving the same qualitative pattern: path-type JS remains much larger than label JS, and the human-minus-ChatGPT H_2 gap remains positive. Therefore, the headline result is not that every path-based statistic is invariant to corpus size, but that the marginal/process gap persists under the available controls. Furthermore, not all diagnostics are equally stable. In HC3, ChatGPT has higher SPG_1, suggesting stronger local predictability gain. However, this result does not replicate cleanly in raw M4 and becomes small under M4 length controls. Rare-path mass is also corpus-sensitive and does not support a simple claim that human text always has a richer syntactic long tail. We therefore treat SPG_k and rare-path mass as secondary diagnostics.

6 Discussion

6.1 Syntactic ingredients versus syntactic composition

The results support the distinction between syntactic ingredients and syntactic composition. Marginal dependency-label distributions describe which grammatical relations a corpus uses, while transition entropy and path-type distributions describe how those relations are assembled along dependency-tree paths. Across HC3 and M4, human and ChatGPT text are relatively close in the first sense but differ in the second.

This is how generated text may exhibit syntactic process regularization. Lower transition entropy does not imply that generated text is ungrammatical, simplistic, or lower quality, but just means adjacent dependency relations along paths are more predictable. Likewise, larger path-type divergence does not mean that generated text uses invalid syntax.

6.2 Robust and corpus-sensitive diagnostics

The experiments also show that process-level diagnostics differ in stability. SPG_k is a useful diagnostic because it asks how much local syntactic context reduces uncertainty about the next dependency relation. However, it combines marginal and conditional entropy, so changes in either component can affect the score. In our experiments, SPG_1 reveals a strong HC3 effect but is less stable in M4 and under length controls. We therefore interpret SPG_k as a secondary diagnostic of corpus-specific predictability structure rather than as the primary cross-corpus statistic.

By contrast, $H_2 = H(R_{t+1} \mid R_t)$ directly measures local uncertainty in dependency composition. The H_2 gap replicates across HC3 and M4 and remains positive under the M4 length, depth, and token-budget controls. We therefore use H_2 together with path-type JS divergence as the primary evidence for the marginal-versus-process distinction, while reporting SPG_k and rare-path mass as

informative but less stable diagnostics.

6.3 Relationship to AI-text detection

Although dependency-process statistics may contain signal useful for AI-text detection, detection is not the goal of this paper. Generated-text detectors can exploit many artifacts correlated with source, including topic, formatting, length, and model-specific style, and prior work has raised concerns about the robustness and reliability of AI-generated text detection in realistic settings (Sadasivan et al., 2023; Wu et al., 2025). Our analysis instead asks whether human and LLM-generated text differ in dependency composition after marginal syntactic inventories and length effects are considered.

Therefore, the results should not be read as evidence that transition entropy alone can reliably identify AI-generated text in arbitrary settings. They show that, in the evaluated corpora, generated text differs from human text at a syntactic-process level that is not captured by dependency-label marginals alone.

6.4 Implications for synthetic data and domain regeneration

The findings suggest that generated-text evaluation should look beyond lexical overlap, semantic adequacy, factuality, and broad fluency. These dimensions are important, but they do not measure whether generated text reproduces the distributional structure of human syntax. Dependency-process statistics provide a complementary lens, revealing structural differences even when generated text is fluent and uses a familiar syntactic inventory.

This is especially relevant for synthetic data augmentation, where generated corpora can affect downstream training distributions (Guo et al., 2024). If generated text is used to enlarge a training corpus, then matching topics, labels, or coarse style may not be sufficient. A synthetic corpus could preserve the marginal frequency of dependency labels while changing the transition process by which those labels compose. Downstream models trained on such data may therefore learn a syntactic prior that differs from the human corpus the synthetic data was intended to approximate. SynPro gives a direct way to audit this possibility at the corpus level.

The same issue arises in domain regeneration, where prior work asks whether LLMs can reproduce the syntactic properties of human text domains (Ju et al., 2025). Benchmarks such as M4 also study machine-generated text across domains and generators (Wang et al., 2024). A model may match broad genre conventions while still producing a different distribution over dependency-path templates. Here, process-level syntax can serve as an additional diagnostic alongside semantic similarity, factuality, register, and lexical diversity. It is not a replacement for these measures, but it captures a different mode of failure.

Finally, the corpus-level framing is important. SynPro is better viewed as a measurement tool than as a detector, since its statistics are designed to compare distributions over groups of texts, and not to make high-stakes judgments about individual documents. This makes the framework useful for model evaluation, data auditing, and controlled generation studies, while avoiding the stronger claim that dependency-process features alone can identify authorship for arbitrary texts.

7 Conclusion

We introduced SynPro, a process-level framework for comparing human and LLM-generated syntax. Instead of only counting dependency labels, SynPro extracts dependency-label paths and compares marginal entropy, transition entropy, and distributional divergence over labels, transitions, and full paths. Across HC3 and M4, human and ChatGPT text are relatively close in dependency-label marginals but differ in local transition entropy and root-to-leaf path distributions. In M4, these differences persist under sentence-length matching, length-and-depth matching, and paired token-budget truncation.

These results support a difference between syntactic ingredients and syntactic composition. LLM-generated text may use a familiar inventory of dependency relations while arranging those relations according to a different process. More broadly, syntactic evaluation of generated text should look beyond marginal feature distributions and measure how linguistic structures compose.

8 Limitations

This study has several limitations. First, all experiments are in English and use ChatGPT-generated text. The results may not generalize to other languages, model families, decoding strategies, or instruction-tuning regimes. Extending the analysis to additional generators and multilingual corpora is an important next step.

Second, the analysis depends on automatic dependency parsing. We use spaCy `en_core_web_sm`, whose labels follow spaCy's ClearNLP-derived dependency inventory rather than a pure Universal Dependencies conversion. Parser errors or parser-specific label conventions could influence the estimated path distributions. A stronger follow-up would rerun the analysis with additional parsers or UD-converted parses.

Third, root-to-leaf dependency-label paths are only one way to define a syntactic process over a tree. Other choices, such as bounded local paths, all ancestor-descendant paths, or random walks, may capture different structural properties. We use root-to-leaf paths because they are simple and interpretable, but path-extraction choices should be tested more broadly.

Fourth, although the M4 controls show that the transition-entropy and path-divergence effects survive sentence-length matching, joint length-and-depth matching, and paired token-budget truncation, length still explains part of the raw M4 gap. The controlled results should therefore be treated as the main evidence, rather than the unadjusted M4 comparison alone.

Finally, lower transition entropy should not be interpreted as lower writing quality. Predictability is a structural statistic, not a normative judgment. Generated text can be fluent and useful while still differing from human text in dependency-transition structure.

9 Ethical Considerations

This work analyzes aggregate syntactic differences between human and LLM-generated text. Although dependency-process features could be used in AI-text detection systems, our goal is not that SynPro be used for classification. We caution against using transition entropy or dependency-path statistics as standalone evidence that a particular text was machine-generated, especially in high-stakes settings such as education, employment, or moderation. The analysis uses existing benchmark corpora and operates on dependency labels and aggregate statistics rather than identifying individual authors.

References

Sara Ahmed and Tracy Hammond. 2026. DependencyAI: Detecting AI generated text through dependency parsing. *Preprint*, arXiv:2602.15514.

Danqi Chen and Christopher D. Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 740–750. Association for Computational Linguistics.

Thomas M. Cover and Joy A. Thomas. 2006. *Elements of Information Theory*, 2 edition. Wiley.

Juan De Gregorio, Raúl Toral, and David Sánchez. 2024. Exploring language relations through syntactic distances and geographic proximity. *EPJ Data Science*, 13(61).

Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation*.

Marie-Catherine de Marneffe, Christopher D. Manning, Joakim Nivre, and Daniel Zeman. 2021. Universal dependencies. *Computational Linguistics*, 47(2):255–308.

Richard Futrell, Roger P. Levy, and Edward Gibson. 2020. Dependency locality as an explanatory principle for word order. *Language*, 96(2):371–412.

Richard Futrell, Kyle Mahowald, and Edward Gibson. 2015a. Large-scale evidence of dependency length minimization in 37 languages. *Proceedings of the National Academy of Sciences*, 112(33):10336–10341.

Richard Futrell, Kyle Mahowald, and Edward Gibson. 2015b. Quantifying word order freedom in dependency corpora. In *Proceedings of the Third International Conference on Dependency Linguistics (Depling 2015)*, pages 91–100, Uppsala, Sweden. Uppsala University.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander M. Rush. 2019. GLTR: Statistical detection and visualization of generated text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 111–116. Association for Computational Linguistics.

Dmitriy Genzel and Eugene Charniak. 2002. Entropy rate constancy in text. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 199–206, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. 2023. How close is ChatGPT to human experts? comparison corpus, evaluation, and detection. *Preprint*, arXiv:2301.07597.

Yanzhu Guo, Guokan Shang, Michalis Vazirgiannis, and Chloé Clavel. 2024. Training language models on synthetic text. In *Findings of the Association for Computational Linguistics: NAACL 2024*. Association for Computational Linguistics.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength natural language processing in python.

Daphne Ippolito, Daniel Duckworth, Chris Callison-Burch, and Douglas Eck. 2020. Automatic detection of generated text is easiest when humans are fooled. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1808–1822. Association for Computational Linguistics.

Yingqi Jing, Paul Widmer, and Balthasar Bickel. 2021. Word order variation is partially constrained by syntactic complexity. *Cognitive Science*, 45(11):e13056.

Da Ju, Hagen Blix, and Adina Williams. 2025. Domain regeneration: How well do LLMs match syntactic properties of text domains? In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 2367–2388, Vienna, Austria. Association for Computational Linguistics.

Eliyahu Kiperwasser and Yoav Goldberg. 2016. Simple and accurate dependency parsing using bidirectional LSTM feature representations. *Transactions of the Association for Computational Linguistics*, 4:313–327.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. 2023. A watermark for large language models. In *Proceedings of the 40th International Conference on Machine Learning*, pages 17061–17084. PMLR.

Yafu Li, Qintong Li, Leyang Cui, Wei Bi, Zhilin Wang, Longyue Wang, Linyi Yang, Shuming Shi, and Yue Zhang. 2024. MAGE: Machine-generated text detection in the wild. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 36–53. Association for Computational Linguistics.

Haitao Liu. 2008. Dependency distance as a metric of language comprehension difficulty. *Journal of Cognitive Science*, 9(2):159–191.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D. Manning, and Chelsea Finn. 2023. DetectGPT: Zero-shot machine-generated text detection using probability curvature. In *Proceedings of the 40th International Conference on Machine Learning*, pages 24950–24962. PMLR.

Fermín Moscoso del Prado Martín. 2025. Measuring grammatical diversity from small corpora: Derivational entropy rates, mean length of utterances, and annotation invariance. *Computational Linguistics*, 51(4):1191–1233.

Alberto Muñoz-Ortiz, Carlos Gómez-Rodríguez, and David Vilares. 2024. Contrasting linguistic patterns in human and LLM-generated news text. *Artificial Intelligence Review*, 57(10):265.

Joakim Nivre. 2008. Algorithms for deterministic incremental dependency parsing. *Computational Linguistics*, 34(4):513–553.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajič, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. 2020. Universal dependencies v2: An evergrowing multilingual treebank collection. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4034–4043, Marseille, France. European Language Resources Association.

Vinu Sankar Sadasivan, Aounon Kumar, Sriram Balasubramanian, Wenxiao Wang, and Soheil Feizi. 2023. Can AI-generated text be reliably detected? In *Proceedings of the 40th International Conference on Machine Learning*. PMLR.

Claude E. Shannon and Warren Weaver. 1949. *The Mathematical Theory of Communication*. University of Illinois Press.

Dhruv Verma, Nicholas Tomlin, and Dan Klein. 2023. Revisiting entropy rate constancy in text. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 14677–14694, Singapore. Association for Computational Linguistics.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Toru Sasaki, Thomas Arnold, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024. 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.

Jinyan Wu, Shu Yang, Runzhe Zhan, Yifan Yuan, Derek F. Wong, and Lidia S. Chao. 2025. A survey on LLM-generated text detection: Necessity, methods, and future directions. *Computational Linguistics*, 51(1):275–338.

Olga Zamaraeva, Dan Flickinger, Francis Bond, and Carlos Gómez-Rodríguez. 2025. Comparing LLM-generated and human-authored news text using formal syntactic theory. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9041–9060, Vienna, Austria. Association for Computational Linguistics.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Defending against neural fake news. In *Advances in Neural Information Processing Systems*, volume 32.