ChatGPT

- Sec.1 (p.1): “accuracy can be misleading, and we should address a different question: can an automatic type migration tool produce code that passes the TypeScript type checker?” [1] 既存手法は個々の型注釈予測の高精度を報告しているが、本論文ではそれが誤解を招く可能性があると指摘している [1] 。つまり、単なる予測精度ではなく、生成されたコードが実際に型チェックを通るかどうかを評価すべきである。（失敗モード1）
- Sec.3.3 (p.10): “we use a process we call type weaving to combine type predictions with the original JavaScript code.” [2] TypeWeaverの統合手順では、予測された型注釈を元のJavaScriptコードに「編み込む（weaving）」ことでTypeScriptコードを生成するプロセスが明示されている [2] 。また、CommonJSからESモジュールへの変換ステップも型チェック成功率に影響することが示唆されている。（失敗モード2）
- Sec.3 (Rejecting non-type predictions, p.3): “These predictions need to be rejected or cleaned for type prediction to work.” [3] 型注釈予測モデルはしばしば構文的に無効なトークン列（型ではない出力）を生成するため、それらの予測値は型付けのために除外・クリーニングする必要があると述べている [3] 。無効な予測が混入すると型チェック評価が破壊される。（失敗モード3）

引用キー案: `typeweaver` — Ming-Ho Yee and Arjun Guha, 2023, ECOOP (European Conference on Object-Oriented Programming).

---

[1] [2] [3] Do Machine Learning Models Produce TypeScript Types That Type Check?
https://drops.dagstuhl.de/storage/00lipics/lipics-vol263-ecoop2023/LIPIcs.ECOOP.2023.37/LIPIcs.ECOOP.2023.37.pdf