Perfect — you successfully ran the script, and I now have what I need to guide the next steps.

**✅ Key Findings from linie\_mit\_polygon.csv**

Based on your terminal output and the script:

| **Aspect** | **Description** |
| --- | --- |
| **Rows** | 3,129 |
| **Columns** | 11 fields, including geometry, line info, and station names |
| **Geometry** | Field "Geo shape" contains LineString geometries (correct format for SUMO) |
| **Station Names** | "START\_OP.1" and "END\_OP.1" fields contain readable names |
| **Edge Metadata** | Includes track gauge ("TRACK GAUGE"), start/end km, and line ID |

**🔍 Observations**

1. **Well-structured LineStrings** in the Geo shape column: good basis for defining SUMO edges with geometry.
2. **Named endpoints (START\_OP.1 & END\_OP.1)**: useful for edge naming, logging, and GTFS linking.
3. **Line grouping (Linie, Line)**: helpful for grouping or filtering routes later.
4. **CRS** is WGS84 (lon/lat) — no need for projection conversion.
5. **Usability**: This file is excellent for creating a lightweight, **segment-based SUMO network**, where:
   * Each row = 1 edge in SUMO.
   * Nodes = endpoints of each LineString (can be generated on-the-fly).
   * We can name edges like START\_END\_lineid for readability.

**🛤️ Next Step: Define Network Construction Strategy**

Before jumping into code, would you like a **step-by-step logic plan** for building a SUMO .net.xml using this file? This would include:

* How to extract nodes from LineString endpoints
* How to construct readable edge IDs
* How to handle duplicates or overlapping geometries
* How to organize the output directory structure

Would you like that next?