**🧠 PhD Thesis Summary – *"Analysis of Virtual Coupling in Operations Through Railway Networks"***

**🎯 Overall Goal:**

To **simulate and analyze Virtual Coupling (VC)** in a real, partially upgraded railway network (Swiss-based), assessing operational and emergency scenarios using **SUMO + Python**.

**⏳ Timeline & Milestones**

| **Milestone** | **Deadline** | **Description** |
| --- | --- | --- |
| **Progress Report 1** | 30 June 2025 | Real network simulation, no VC yet. KPI logging system built. |
| **Progress Report 2 + Paper** | 31 December 2025 | VC simulations (scenarios + emergencies), results → journal paper draft |
| **Progress Report 3** | 30 June 2026 | Full VC scenario set, extensive testing, advanced ML/logic optional |
| **Progress Report 4 (Final)** | 31 December 2026 | Final evaluation, thesis writing, defense prep |
| **PhD Thesis Submission** | 31 December 2026 | Submit final PhD + published journal paper (mandatory at your university) |

**🧩 Phase-Based Work Structure**

**✅ Phase 1 — Baseline (Now–June 2025)**

1. **Swiss Railway Network Rebuild**
   * Clean topology in SUMO from swissTNE\_Base\_20240507.gpkg
   * Fully parse GTFS, jahresformation, rollmaterial, etc.
2. **Route & Vehicle Metadata System**
   * Build 3 supporting data files:
     + route\_vehicle\_info.csv – Route ↔ vehicle mapping
     + vehicle\_type\_info.csv – SUMO-ready parameters for all vehicles
     + route\_edge\_mapping.csv – Origin, destination, intermediate stops + edge sequences
3. **.rou.xml Creation System**
   * Python script to generate SUMO routes based on the 3 files above
   * Includes randomization of parameters (dwell time, accel, decel)
4. **Simulation Engine for Non-VC**
   * TraCI-controlled Python loop to simulate normal train operations
   * Export and log KPIs
5. **KPI Logging and Visualization**
   * Delay, headway, dwell time, throughput, edge utilization, emissions
   * Logging to CSV, visualization in matplotlib / seaborn
6. **Start VC Code Skeleton**
   * Modular Python logic for later VC simulation (but not used in this phase)

**✅ Phase 2 — VC Simulation Core (July–Dec 2025)**

1. **VC Operational Scenarios**
   * Simulate networks with:
     + VC vehicles (0%–100%)
     + VC infrastructure (0%–100%)
     + Mixed conditions
2. **VC Emergency Scenarios**
   * Communication failure
   * Emergency braking
   * Track obstruction
3. **VC Logic Engine (Python)**
   * train\_manager.py: coupling logic
   * scenario\_generator.py: auto-mix VC/legacy vehicles
   * emergency\_handler.py: inject faults
   * data\_logger.py: track KPIs
   * analyzer.py: parse SUMO outputs
4. **Paper Drafting**
   * Paper is based on Progress Reports 1 + 2
   * Publish in a journal by Dec 2025

**📊 KPI Framework**

| **Category** | **KPIs Tracked** |
| --- | --- |
| **Operational** | Delay per train, travel time, headway, train throughput |
| **Energy & Emissions** | Estimated fuel/electricity use, emission output |
| **Capacity & Usage** | Segment utilization, track bottlenecks, junction throughput |
| **Safety** | VC failure rates, braking violations, emergency resolution stats |
| **VC Performance** | Coupling success rate, average VC train length, disruption impact |

**🧠 Research Contribution Highlights**

| **Gap in Literature** | **Your Contribution** |
| --- | --- |
| Unrealistic full VC assumptions | Simulate **mixed VC and legacy** trains and infrastructure |
| No network-scale KPI evaluations | System-wide KPI logging and comparison |
| Pairwise VC only | Route-level and network-level VC behavior |
| No emergency fallback scenarios | **Model emergencies and response logic** |
| Complex, overkill ML | Optional use of **lightweight ML** (DQN, trees, logistic) |

**🧰 Technical Strategy**

* **Simulation Tool**: SUMO + Python (TraCI)
* **Data Source**: SwissTNE, GTFS, SBB logs, rollmaterial
* **Scripts & Files**:
  + SUMO input generators (edges, nodes, con)
  + .rou.xml generator with randomized realism
  + KPI logger, VC control logic
* **Analysis Tools**: pandas, seaborn, matplotlib
* **Optional ML**: scikit-learn, TensorFlow-lite (for VC decisions)
* **VC Tags**: Used to label vehicles and track segments

**🛡️ Escape Plan (Career Strategy)**

* Show **system-level thinking** (beyond algorithms)
* Prove **transport operations + simulation expertise**
* Develop skills German employers need:
  + **Python**, **GTFS**, **KPIs**, **SUMO**, **mixed operations**
  + Safety culture (influenced by your aviation background)
* Modular code portfolio = immediate value to transport authorities or firms (Deutsche Bahn, Siemens, etc.)

As i want to use this PhD thesis as a showcase for the possible future german employers in the field trnasport planner/transport analyzer/transport engineer positions (private or public companies or institutions), I want to research ans simulate both meaningful and useful KPIs for industry and also for academy (for academic paper). And also even though now we are simulating Swiss network so far, in the later stages i want to simulate German railways too (i don't have any datasets yet) in order to maximize my chances of getting a job in Germany in the mentioned positions.

Great — this is a **strategic move** that aligns your PhD thesis with both **academic publication standards** *and* the **practical expectations of German transport employers** (e.g., DB Netz, Deutsche Bahn, Fraunhofer IVI, city planning offices, Siemens Mobility, RMV, etc.). Here's a full **framework for KPIs and future extensions**, tailored to both academic contribution and industry relevance — and aligned with your escape plan:

**🎯 KPI Strategy: Dual Track (Academic + Industry)**

| **KPI Type** | **Why It Matters (Academia)** | **Why It Matters (Industry & Planning)** |
| --- | --- | --- |
| **Delay** (per train, avg, total) | Standard metric in railway performance studies | Operational reliability and timetable robustness |
| **Headway** (min, avg) | Assesses capacity efficiency under VC | Determines track capacity and service frequency |
| **Throughput** (trains/hour/segment) | System-level capacity utilization | Infrastructure bottleneck analysis and line planning |
| **Emissions** (CO₂/train, total) | Environmental benefit of VC, sustainability argument | Carbon reduction reporting, energy optimization |
| **Energy Use** (kWh/km) | Useful for modeling eco-driving, power use | Infrastructure electrification planning |
| **Coupling Success Rate** | VC stability, success under mixed conditions | Realistic implementation feasibility |
| **Safety Violations** (e.g., braking distance not met) | Simulation of fallback/emergency behavior | Safety case validation, risk mitigation planning |
| **Occupancy Simulation** (optional) | Not common in SUMO, but great for extensions | Supports train size planning, peak analysis |
| **Junction Delay** | Shows complexity at high-load points | Node design, reconfiguration justification |

**📈 KPI Implementation Plan (Phases)**

**🔹 Phase 1 — Non-VC Baseline (Progress Report 1)**

| **KPI** | **Tools** | **Data Source** |
| --- | --- | --- |
| Delay per train | SUMO tripinfo.xml, TraCI logging | GTFS schedule vs actual simulation times |
| Headway | Custom tracker in Python | stop\_times.txt, TraCI vehicle arrivals |
| Throughput | Count trains passing edges/junctions | TraCI edge logs |
| Emissions / Energy | emissions.xml, SUMO emission model | Vehicle types (from your enriched data) |
| Travel Time | Start–end duration from tripinfo | Simulated logs |

🧠 **Objective**: Create baseline benchmark → "this is current performance with no VC".

**🔹 Phase 2 — VC + Emergencies (Progress Report 2)**

| **KPI** | **Additional Logic Needed** | **Data Source** |
| --- | --- | --- |
| VC Coupling Frequency | train\_manager.py state transitions | Your own Python logic |
| VC Failure Rate | Emergency events logged in TraCI | emergency\_handler.py logs |
| Safety Margin Violations | Headway monitoring during VC | TraCI vehicle.getDistance checks |
| De-coupling Events Logged | Controlled vs forced decoupling tracking | Your own events + scenario flags |
| Delay Recovery After Fault | Compare delay slopes pre/post event | Custom log processor |

🧠 **Objective**: Show how VC impacts safety, delay, and energy under imperfect conditions.

**🔄 Reusability for German Employers**

1. **Industry-Ready KPIs**: DB, Siemens, Thales, and others rely on:
   * Delay statistics per station and segment
   * Junction-level congestion
   * Emission reductions per timetable scenario
   * Capacity improvement (before vs after upgrade)
   * Reliability/resilience under disruption
2. **Visualization Plan**:
   * KPI dashboards in matplotlib / seaborn
   * Before/after VC comparison plots
   * Timeline of coupling events, delays, energy
3. **Documentation & Reporting**:
   * CSV exports for every simulation
   * Short summary reports per batch run
   * Modular simulation batches by scenario
   * One-click rerun script for demo purposes (important for portfolio use)

**🧩 Future Extensions (Germany)**

You can reuse your **entire architecture**:

* route-vehicle-info, vehicle\_type\_info, route-edge-mapping, .rou.xml generator, VC module, KPI logger

Once you obtain:

* German GTFS (from VBB, DB, MVV, VRR, etc.)
* German open rail topology (Deutsche Bahn is moving datasets to [Mobilithek](https://mobilithek.info) + GovData)
* DB station lists, line assignments (like your Swiss jahresformation)

🎯 **Your benefit**: no need to re-code logic — just swap in datasets + remap stations to edges.

**🗺️ Long-Term Simulation Roadmap**

| **Phase** | **System** | **Features** |
| --- | --- | --- |
| P1 | Swiss network baseline | SUMO, no VC, realistic timetables |
| P2 | Swiss network w/ VC | Partial VC trains + infra, fault simulation |
| P3 | German regional sample | Same framework with DB data (when acquired) |
| P4 | Compare 🇨🇭 vs 🇩🇪 results | KPIs, operational/infra differences |
| P5 | Resume-wide batch testing | Dozens of simulation combinations for academic paper + hiring showcase |