Driving InsurTech Innovation with Real-World Insurance Data: From ML to AI

Real-world Usecases

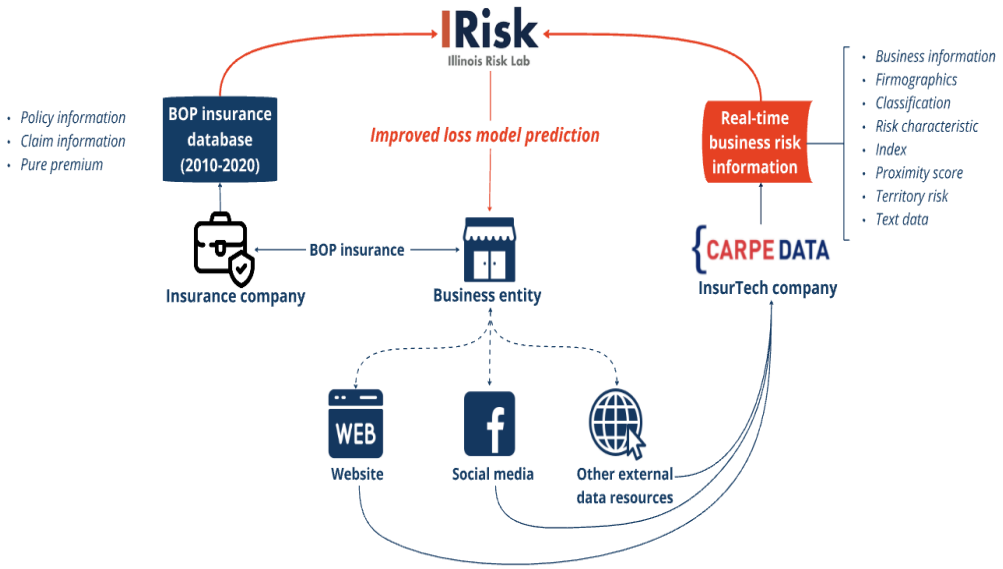
Zhiyu (Frank) Quan

University of Illinois Urbana-Champaign

Overview

- InsurTech Innovation
- Data is the New Oil: Federated Learning
- Automated Machine Learning
- Agentic AI: Claim Automation
- AI is neither a Hype nor Magic

InsurTech Innovations with Real-World Data

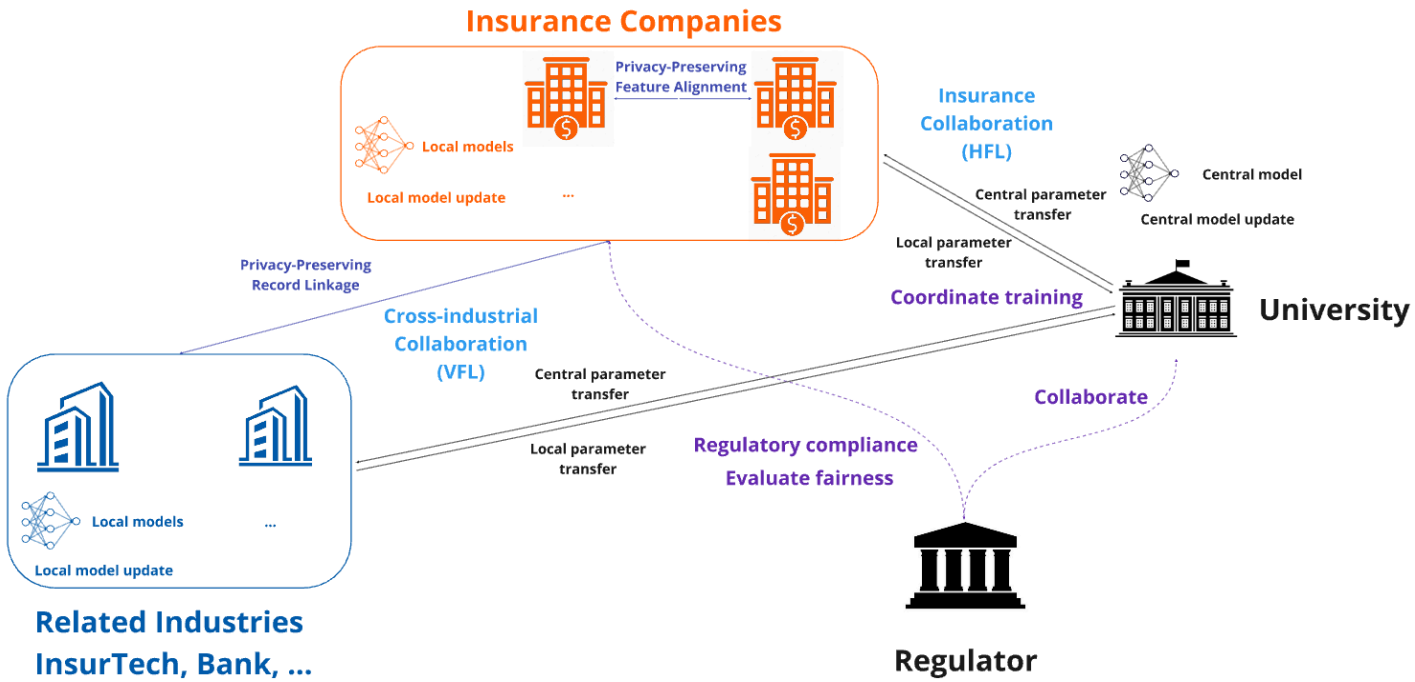


- The evolving risk landscape calls for ***additional*** predictive factors beyond ***traditional*** rating variables to achieve more accurate pricing[1].
- The ***university*** serves as an independent third-party evaluator for the InsurTech data vendor, quantifying the predictive lift achieved when insurers incorporate additional predictive factors into their in house models.

[1] Quan, Z., Hu, C., Dong, P., & Valdez, E. A. (2025). Improving business insurance loss models by leveraging InsurTech innovation. *North American Actuarial Journal*, 29(2), 247-274.

Federated Learning: Facilitates Industry-wide Partnership

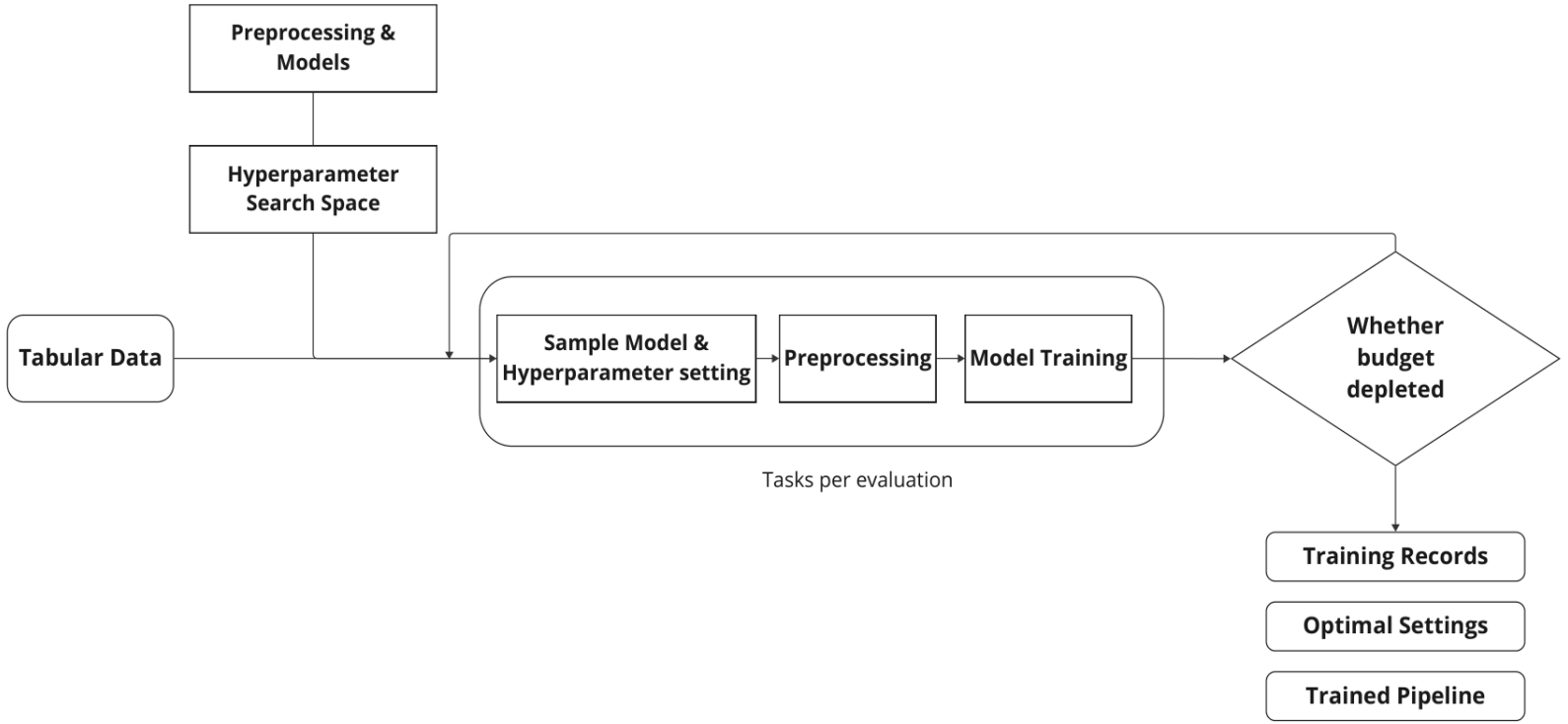
- Can we solve industry-wide ***pain points*** while preserving ***data privacy***?[2]
- Even large carriers can gain valuable insights from data contributed by smaller insurers.



[2] Dong, P., Feng, R., Quan, Z., & Wang, T. (2026). Bridging the Divide While Walking a Tightrope: Evidence from the Insurance Industry on Federated Data Sharing.

Automated Machine Learning

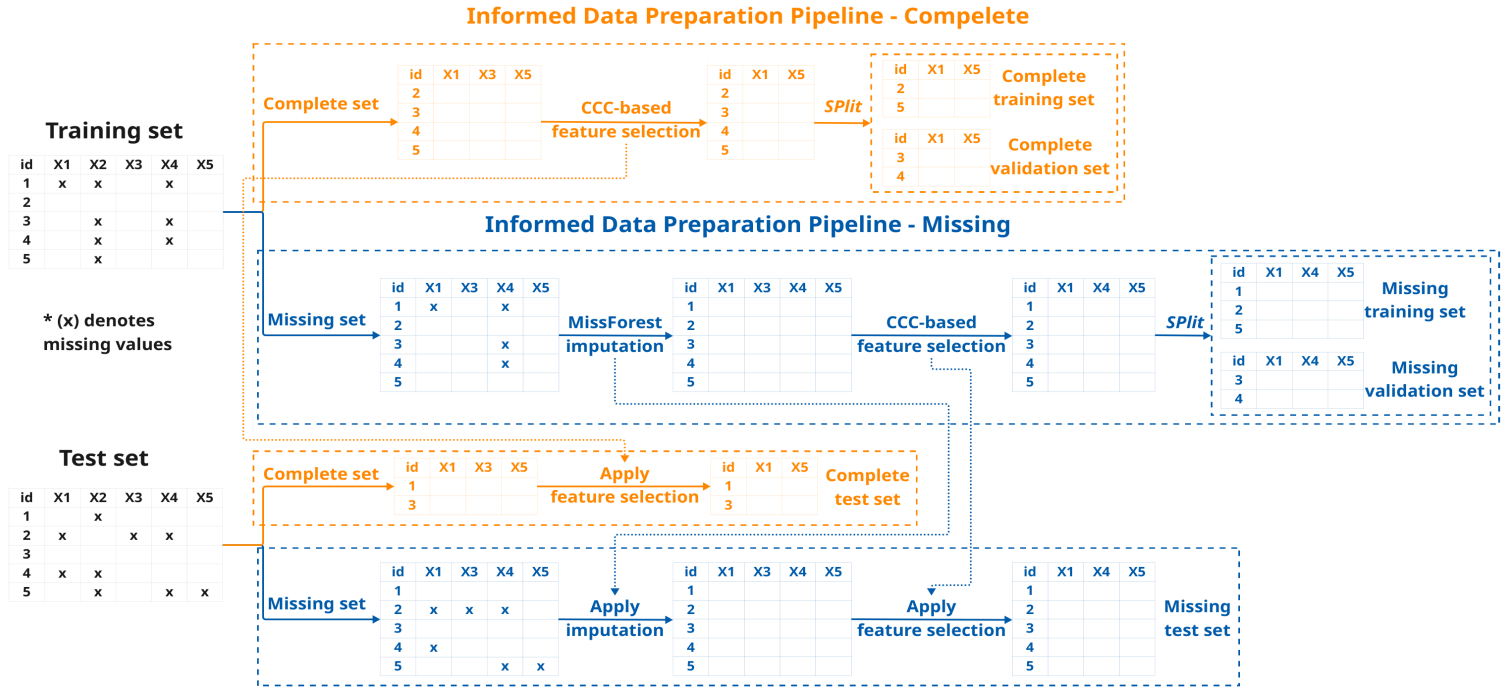
- ***Empower*** actuaries to unlock the power of data science through AutoML[3].



[3] Dong, P., & Quan, Z. (2025). Automated machine learning in insurance. *Insurance: Mathematics and Economics*, 120, 17-41.

- Do we still need data scientists/machine learning researchers in the AI era? ***Domain knowledge*** and ability to transform business problem into optimization frameworks are more important than ever[4].

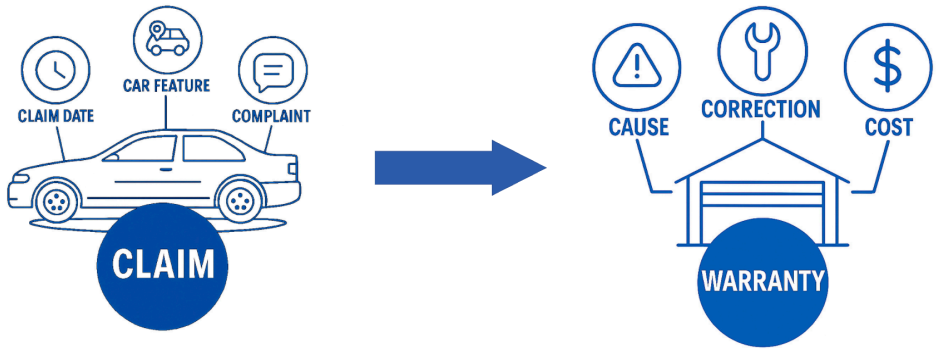
- Interested to try out[5]?



[4] Guo, J., Dong, P., & Quan, Z. (2026). Starting Off on the Wrong Foot: Pitfalls in Data Preparation.

[5] https://github.com/PanyiDong/InsurAutoML

Claim Agentic AI

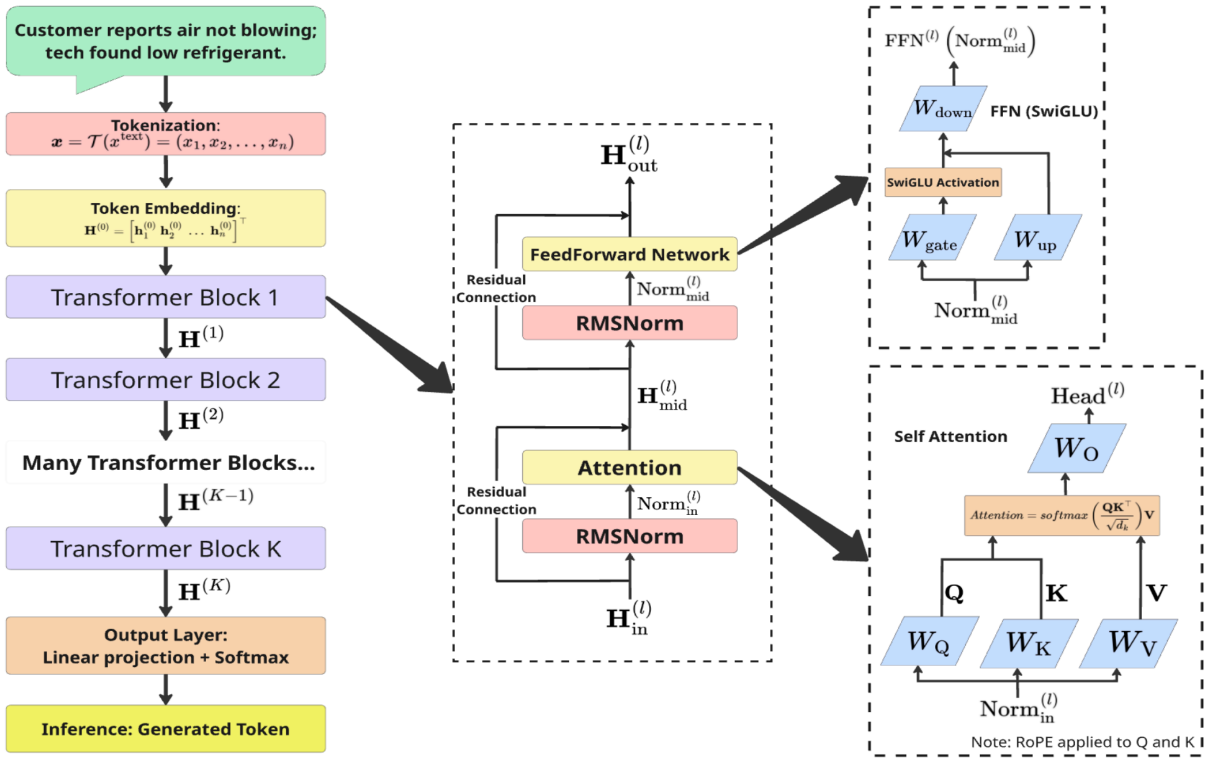


- Customer **complaint** descriptions — free-text narratives of vehicle issues.
- Diagnosed **causes** — brief technician notes or internal codes.
- **Corrective actions** — usually free-text descriptions of repairs.
- **Loss codes** — standardized categorizations of claim types.

- Large Language Models (LLMs) can ***assist*** claim adjusters, service writers, and customers by quickly and accurately predicting repair needs—parts, labor, and cost[6].

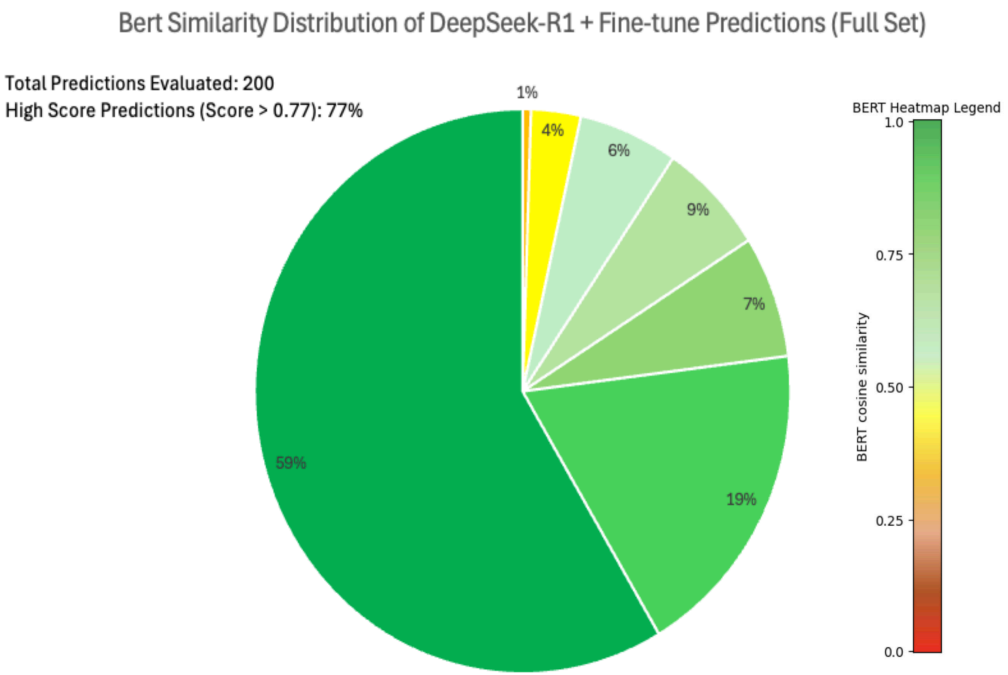
[6] Mo, Z., Quan, Z., O'Donohue, E., & Zhong, K. (2026). Claim Automation using Large Language Model.

Localized Open-Source LLM



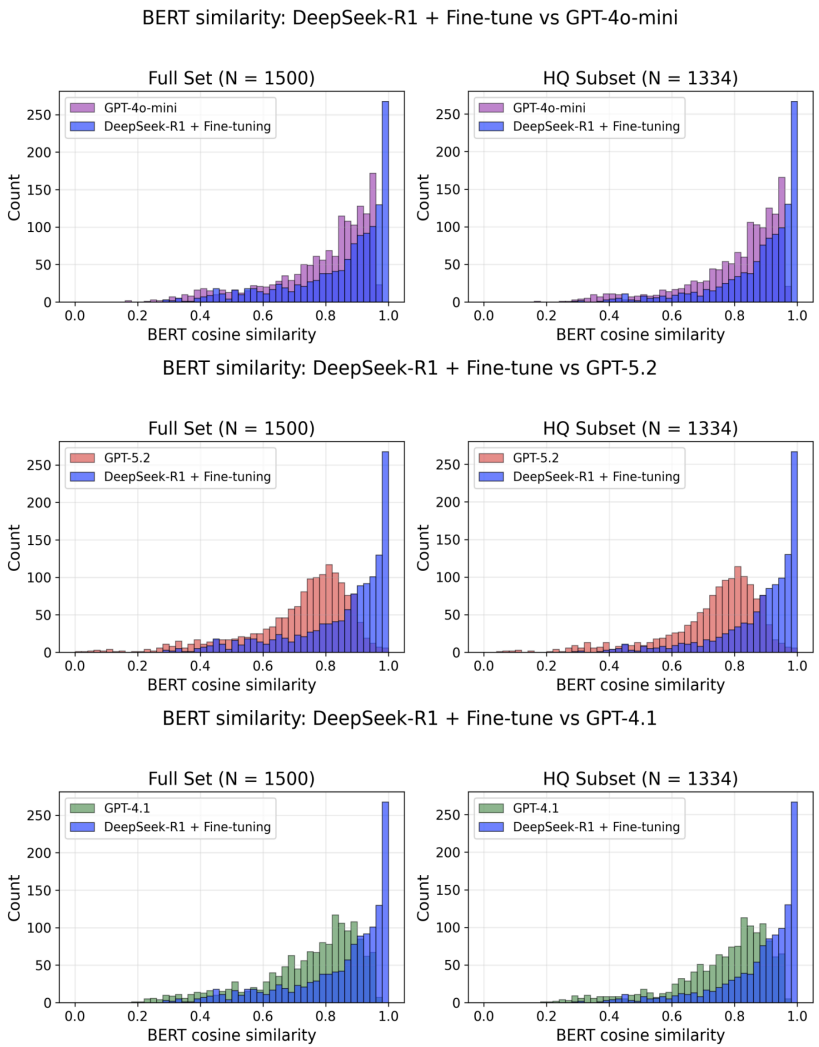
- LLMs can be ***locally*** deployed on GPUs on-prem. We fine-tune an ***open-source*** model (DeepSeek-R1-Distill-Llama-8B) on an 8×A100 GPU cluster with ***9 millions*** historical claim records.

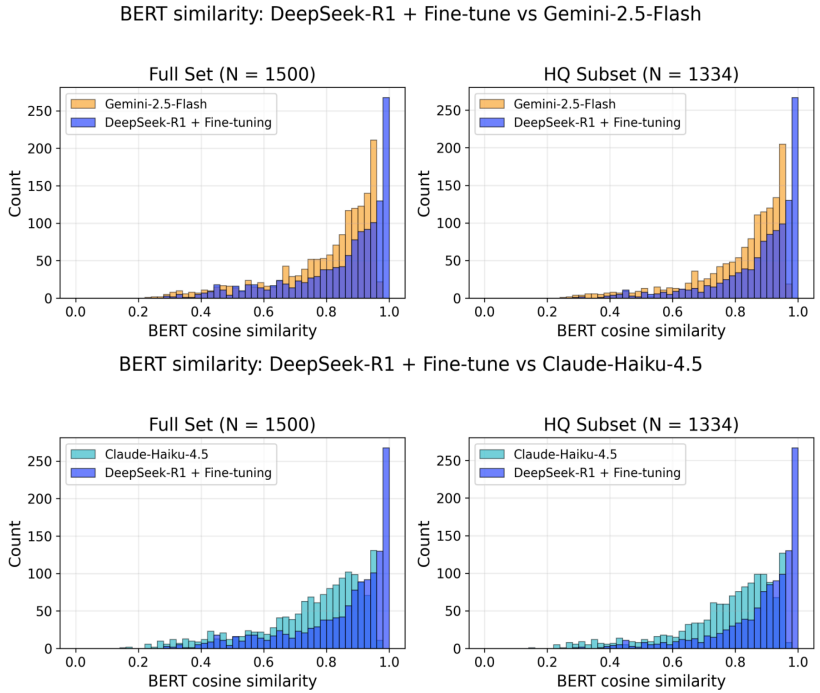
Agent Handle Routine Claims



- Human and automated evaluations on model prediction:
  - **85.8% of predictions scored above 0.77**, meaning they are semantically correct with only minor wording differences.
  - Most remaining predictions fall between 0.4 and 0.8, reflecting partial correctness.
  - **A small portion (~2%) scored below 0.3**, indicating occasional misunderstandings or predicting a different corrective action than truth.

Commercial Sate-Of-The-Art LLMs





AI is neither a Hype nor Magic

- LLMs are not designed to be intelligent.
- Insurance data remains siloed.
- Insurers hold a wealth of proprietary data to unlock AI's potential.
- The competitive edge will belong to those who unlock its AI potential first.

Thank you! Q&A

Appendix A. Human Evaluation

Table 1: Human evaluation scoring guidelines used in all annotation tasks.

Score	Meaning	Description
1.0	Perfect Match	The predicted actions fully and precisely cover the ground-truth actions. No missing or incorrect actions.
0.8	Strong Match	The predicted actions mostly cover the ground truth, with only minor omissions or paraphrasing and no major errors.
0.6	Partial Match	The prediction covers some key actions but misses important parts. Still related and somewhat helpful.
0.4	Weak Match	Very limited overlap. Only a few actions (or fragments) are relevant to the ground truth.
0.2	Minimal Match	Prediction is largely irrelevant or incorrect, with almost no valid overlap with the ground truth.
0.0	No Match	Completely unrelated or invalid prediction, with no correspondence to the ground truth.

Appendix B. Automated Evaluation

C Formal Definitions of Automated Evaluation Metrics

In this section, we formally define all automated evaluation metrics used in section 3.4. Let y denote the reference corrective action and $\hat{y}$ the model-predicted action, both represented as natural language strings. All metrics quantify the similarity between $\hat{y}$ and y from different perspectives.

Edit distance similarity. Let $\mathrm{ED}(\hat{y}, y)$ denote the character-level Levenshtein edit distance between $\hat{y}$ and y. We define the normalized edit-distance similarity as

$$\mathrm{Sim}_{\mathrm{ED}}(\hat{y}, y) = 1 - \frac{\mathrm{ED}(\hat{y}, y)}{\max\left(|\hat{y}|, |y|\right)}, \tag{C.1}$$

where $|\cdot|$ denotes string length (number of characters).

N-gram precision. Let $\mathcal{G}_n(s)$ denote the multiset of n-grams (tokens) extracted from a string s, and let $c_s(g)$ denote the count of n-gram g in s. We define n-gram precision as

$$\mathrm{Prec}_n(\hat{y}, y) = \frac{\sum_{g \in \mathcal{G}_n(\hat{y})} \min\left(c_{\hat{y}}(g),\, c_y(g)\right)}{\sum_{g \in \mathcal{G}_n(\hat{y})} c_{\hat{y}}(g)}. \tag{C.2}$$

Sentence BLEU. We adopt sentence-level BLEU as implemented in sacreBLEU (Post, 2018), based on the original BLEU formulation (Papineni et al., 2002). The clipped n-gram precision is defined as

$$p_n(\hat{y}, y) = \frac{\sum_{g \in \mathcal{G}_n(\hat{y})} \min\left(c_{\hat{y}}(g),\, c_y(g)\right)}{\sum_{g \in \mathcal{G}_n(\hat{y})} c_{\hat{y}}(g)}. \tag{C.3}$$

Sentence BLEU is computed as the geometric mean of modified n-gram precisions with a brevity penalty:

$$\mathrm{BLEU}(\hat{y}, y) = \mathrm{BP} \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n(\hat{y}, y)\right), \tag{C.4}$$

where w_n are uniform weights, $N = 4$ in our experiments, and the brevity penalty is

$$\mathrm{BP} = \begin{cases} 1, & |\hat{y}| > |y|, \\ \exp\left(1 - \frac{|y|}{|\hat{y}|}\right), & |\hat{y}| \le |y|. \end{cases} \tag{C.5}$$

TF–IDF cosine similarity. Let $\mathcal{D} = \{y_1, \ldots, y_T, \hat{y}_1, \ldots, \hat{y}_T\}$ denote the combined corpus consisting of all reference and predicted actions. A TF–IDF vectorizer is fitted on $\mathcal{D}$ with n-gram range $(1, 2)$. For any string s, let $\mathbf{v}(s) \in \mathbb{R}^d$ denote its TF–IDF representation under this shared vocabulary and inverse document frequency weighting.

The TF–IDF cosine similarity between a prediction $\hat{y}$ and its reference y is defined as

$$\mathrm{Sim}_{\mathrm{tfidf}}(\hat{y}, y) = \frac{\mathbf{v}(\hat{y})^\top \mathbf{v}(y)}{\|\mathbf{v}(\hat{y})\|_2 \, \|\mathbf{v}(y)\|_2}. \tag{C.6}$$

BERT cosine similarity. We adopt a pretrained sentence embedding model from the Sentence-Transformers framework. Specifically, we use `all-mpnet-base-v2` (Song et al., 2020) to encode each string into a fixed-dimensional vector.

Let $\mathbf{e}(s) \in \mathbb{R}^d$ denote the sentence embedding of a string s produced by the encoder. The BERT cosine similarity between a prediction $\hat{y}$ and its reference y is defined as

$$\mathrm{Sim}_{\mathrm{BERT}}(\hat{y}, y) = \frac{\mathbf{e}(\hat{y})^\top \mathbf{e}(y)}{\|\mathbf{e}(\hat{y})\|_2 \, \|\mathbf{e}(y)\|_2}. \tag{C.7}$$

BERTScore (F1). We use BERTScore as a token-level semantic similarity metric based on contextualized embeddings. Given a prediction $\hat{y}$ and a reference y, both are first tokenized and encoded by a pretrained transformer encoder into contextual token embeddings.

Let $\mathbf{h}_i(\hat{y}) \in \mathbb{R}^d$ denote the embedding of the i-th token in $\hat{y}$, and $\mathbf{h}_j(y) \in \mathbb{R}^d$ the embedding of the j-th token in y. The pairwise token similarity is defined by cosine similarity

$$s_{ij} = \frac{\mathbf{h}_i(\hat{y})^\top \mathbf{h}_j(y)}{\|\mathbf{h}_i(\hat{y})\|_2 \, \|\mathbf{h}_j(y)\|_2}.$$

BERTScore precision and recall are defined as

$$P_{\mathrm{BS}}(\hat{y}, y) = \frac{1}{|\hat{y}|} \sum_i \max_j s_{ij}, \qquad R_{\mathrm{BS}}(\hat{y}, y) = \frac{1}{|y|} \sum_j \max_i s_{ij}. \tag{C.8}$$

The BERTScore F1 is given by

$$\mathrm{BERTScore}_{\mathrm{F1}}(\hat{y}, y) = \frac{2 P_{\mathrm{BS}}(\hat{y}, y) R_{\mathrm{BS}}(\hat{y}, y)}{P_{\mathrm{BS}}(\hat{y}, y) + R_{\mathrm{BS}}(\hat{y}, y)}. \tag{C.9}$$

Appendix B. Automated Evaluation

BLEURT. We adopt BLEURT as a learned neural evaluation model. Formally, let

$$f_{\theta_{\text{BLEURT}}} : \mathcal{U}^* \times \mathcal{U}^* \to \mathbb{R}$$

denote a pretrained neural scoring function that maps a reference action y and a predicted action $\hat{y}$ to a real-valued semantic similarity score.

The BLEURT score is defined as

$$\text{BLEURT}(\hat{y}, y) = f_{\theta_{\text{BLEURT}}}(y, \hat{y}), \tag{C.10}$$

where $f_{\theta_{\text{BLEURT}}}$ is instantiated as a Transformer-based regression model fine-tuned to predict human judgment scores over text pairs. In our experiments, we use the publicly released checkpoint `Elron/bleurt-base-128` hosted on the HuggingFace Model Hub and loaded via the `transformers` library (Wolf et al., 2020). Given a pair $(y, \hat{y})$, the model outputs a single scalar logit, which is used directly as the BLEURT score.

LLM-as-a-Judge score. We implement an LLM-based evaluator to score predicted actions according to a predetermined evaluation rubric.

Formally, let

$$g_\phi : \mathcal{X} \to \mathcal{T}$$

denote a pretrained LLM with fixed parameters ϕ, and let

$$\pi : \mathcal{U}^* \times \mathcal{U}^* \to \mathcal{X}$$

be a deterministic prompting function that maps a prediction–reference pair $(\hat{y}, y)$ into a structured evaluation prompt.

The LLM-as-a-Judge score is defined as

$$\text{Judge}(\hat{y}, y) = \text{Parse}(g_\phi(\pi(\hat{y}, y))), \tag{C.11}$$

where $g_\phi(\pi(\hat{y}, y))$ denotes the model-generated evaluation response, and $\text{Parse}(\cdot)$ extracts a scalar score according to a fixed scoring evaluation rubric. In our experiments, g_ϕ is instantiated as `ChatGPT-4o-mini`, the scoring rubric is defined in Table 2, and all evaluations are performed using deterministic decoding.

Appendix C. Case Study

G Case Study: Repair vs. Replace in Non-Repairable Tire Claims

According to United States. Department of Transportation. National Highway Traffic Safety Administration (2008), punctures located on the tire sidewall are generally considered non-repairable and require tire replacement rather than repair. This rule is well established in automotive safety standards and is routinely enforced in warranty claim operations.

This practice is clearly reflected in our dataset. Among the $2,953$ tire-related claims that contain lexical indicators of sidewall damage (e.g., "inside wall", "in side wall"), only 4 cases resulted repair-type corrective actions (e.g., "repair tire", "patch tire", or "remove nail"), yielding an empirical repair probability of $\frac{4}{2953} \approx 1.36 \times 10^{-3}$. This extreme imbalance closely mirrors real-world repair decisions for sidewall damage. If we assume the repair probability follows a Binomial distribution (not suitable for the imbalance case, again introducing conservative bias), then the expected number of repairs is 4. The fine-tuned model predicts only 1 repair-type actions among the $2,953$ tire-related claims, whereas non-fine-tuned models predict 262 (DeepSeek-R1 + Prompt) and 65 (Qwen-Instruct + Prompt), respectively. These large deviations indicate severe, statistically significant distributional shifts, with non-fine-tuned models systematically overestimating repair probability for practically non-repairable sidewall damage claims.

In contrast, the fine-tuned model's predictions are statistically indistinguishable from the empirical reference distribution. Human evaluation suggests that non-fine-tuned models often rely on generic automotive narratives (e.g., "punctures can often be patched"), reflecting common consumer-level knowledge rather than real-world industry practice. The fine-tuned model, trained on historical warranty claim outcomes, internalizes this highly asymmetric conditional distribution, aligning its prediction with real-world practice.

Appendix D. Research Overview