Reusable Skill Injection: Geometric Motion Plans with Human Motor Expertise

Miroslav David, Karla Stepanova, and Robert Babuska

***Abstract*— Robotic surface-interaction tasks, such as spray painting or welding, require both accurate geometric planning and correct motion execution. While modern motion planners generate valid geometric paths, they often lack the expert motor patterns observed in human operators. Conversely, learning from demonstration often tightly couples execution to the specific training geometry, limiting transferability. We propose a modular framework that decouples geometric motion planning from execution-level expertise. Expert behavior is represented as a vocabulary of interpretable, atomic motor rules, such as velocity scaling and geometric proximity adaptations, that systematically modify a geometrically planned reference path. We utilize a transformer-based model to infer these rule parameters from kinematic trajectory data. We evaluate our approach through dynamic simulation on L-shaped and window-shaped objects, demonstrating that the model successfully extracts execution rules, including behavior conditioned on unobserved CAD features such as structural crossings. This approach allows foundational atomic rules to be transferred to unseen geometries, forming a basis for complex expert heuristics.**

I. INTRODUCTION

Robotic surface-interaction tasks, such as spray painting or welding, require not only accurate geometric motion planning but also appropriate execution strategies. While modern motion planners can generate collision-free paths that satisfy geometric constraints, the resulting trajectories often lack the nuanced motor patterns observed in human expert behavior. Skilled operators adapt velocity, orientation, and tool positioning according to the local geometry of the workpiece, producing execution patterns that are difficult to capture using purely geometric planning methods. An alternative approach is to directly show the desired robot motions through human demonstrations. While such an approach preserves expert motor skills, the resulting trajectories are often tightly coupled to the demonstrated geometry and therefore difficult to transfer to new workpieces.

In this work, we propose a modular approach for integrating human motor expertise with geometric motion planning. Rather than learning full trajectories from demonstrations, expert behavior is represented as a set of interpretable motor rules that modify the execution of a geometrically planned path. These rules describe systematic transformations of the trajectory, such as velocity scaling, spatial offsets, or periodic execution patterns. For example, a rule may increase the traversal speed on straight segments or introduce a consistent offset from the surface to control spray width. The proposed approach aims to decouple geometric planning from execution-level expertise, enabling expert motor strategies to be transferred across different workpiece geometries. A transformer-based model is used to infer rule types and parameters from trajectory data, and the extracted rules are then injected into a physics-based simulation to refine geometrically planned trajectories.

The contributions of this work are: (1) A modular framework for integrating expert motor skills with geometric motion planning through rule-based trajectory refinement. (2) A parametric representation of motor expertise that encodes execution strategies as interpretable trajectory transformation rules.

II. RELATED WORK

Classical approaches to robotic spray painting primarily focus on geometric coverage and physics-based deposition models [1], [2], [3]. A different line of work addresses skill acquisition through learning from demonstrations and policy learning [4], [5], [6]. These methods typically learn a tightly coupled mapping between observed behavior and the task setting in which it is demonstrated. As a result, execution strategies are often tied to the geometry of the demonstrated scenario, limiting transfer to new workpiece shapes or planning contexts.

More recently, data-driven trajectory generation and refinement methods have explored neural approaches to modify or generate robot motion. Residual learning methods augment nominal controllers with learned corrections [7], [8], while transformer-based models have been used to reshape existing trajectories from multimodal inputs such as natural language [9]. In robotic spray painting specifically, recent learning-based planners such as PaintNet and MaskPlanner predict geometric path structures directly from 3D object representations [10], [11]. These methods demonstrate strong generalization in geometric path generation but often rely on implicit latent representations and do not explicitly separate execution-level expertise from the underlying geometric plan.

In contrast, our approach decouples geometric motion planning from execution-level motor expertise. Rather than learning full trajectories end-to-end, we represent expert behavior as a set of interpretable parametric rules that refine a geometric reference path. This modular formulation preserves the transferability of geometric planners while enabling the injection of reusable execution strategies across different object geometries.

III. PROBLEM FORMULATION

Let $\mathcal{T}$ denote the space of all possible trajectories and $\mathcal{O}$ denote the space of all workpiece geometries. Each object geometry $O \in \mathcal{O}$ is represented as a point cloud $O = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_K\}$, where each point $\mathbf{x}_k \in \mathbb{R}^3$. A set of expert demonstrations is defined as $\mathcal{D} = \{(\tau_1, O_1), \ldots, (\tau_n, O_n)\}$. Each trajectory demonstration $\tau_i = \{(t_j, \mathbf{p}_j, \mathbf{q}_j, v_j, w_j)\}_{j=1}^{T_i}$ is a time-indexed sequence of poses consisting of positions $\mathbf{p}_j \in \mathbb{R}^3$, orientations as unit quaternions $\mathbf{q}_j \in \mathbb{S}^3$, linear speeds $v_j \in \mathbb{R}$, and angular speeds $w_j \in \mathbb{R}$.

Given $\mathcal{D}$, our goals are to: (1) extract a set of interpretable rules $\mathcal{R}$ and associated parameter spaces Θ that characterize expert skills relative to the workpiece geometry, (2) learn an inference function $f : \mathcal{T} \times \mathcal{O} \rightarrow \{0,1\}^m \times \Theta$ that identifies active rules, and (3) inject these context-aware skills into a raw geometric trajectory τ^* for a new workpiece O^*. Formally, the skill-enhanced trajectory $\hat{\tau}$ is produced via the composition $\hat{\tau} = (\rho_n(\theta_n, O^*) \circ \cdots \circ \rho_1(\theta_1, O^*))(\tau^*)$, where each rule ρ_k acts as a parametric transformation conditioned on the target object's geometry.

g

IV. METHODOLOGY

This section details the formal representation of rules and the underlying physics governing the trajectory refinement process. An overview of the proposed pipeline is shown in Fig. 1, illustrating the artificial data generation process, rule inference training, and skill transfer to unseen geometries.

A. Rule-Based Skill Representation

Expert skills are modeled as a structured vocabulary of parametric transformations. The raw geometric path τ^* is first decomposed into a sequence of spatial segments $\mathcal{S}(\tau^*) = \{s_1, s_2, \ldots, s_L\}$, where each segment s_i is described by a set of geometric and spatial features c_i (e.g., shape type such as straight or corner, and spatial orientation such as horizontal or vertical).

The trajectory profile for each segment can then be systematically modified by applicable expert rules. A general rule $\rho_k \in \mathcal{R}$ consists of an **activation condition** evaluated against the segment features c_i, and a **dynamic override** parameterized by θ_k. These overrides allow for systematic deviations in velocity or pose without compromising the structural validity of the path.

B. Trajectory Refinement via Dynamic Simulation

To ensure the generated trajectory $\hat{\tau}$ exhibits physically continuous inertial dynamics and avoids instantaneous velocity jumps, the end-effector is modeled as a rigid body. Motion is governed by a pure pursuit tracking controller that determines a target pose at a fixed lookahead distance along the reference path. The controller computes the necessary control forces and torques via proportional feedback, smoothly integrating the dynamically injected velocity and orientation overrides while respecting the underlying kinematic constraints.

C. Data Generation and Preprocessing

Training data are generated using custom L-shaped topologies alongside MaskPlanner window geometries [11] to evaluate generalization across varying surface orientations and structural crossings. Raw geometric paths $(x, y, z, \text{Euler angles}, \texttt{partID})$ are segmented by $\texttt{partID}$, and the orientations are converted to unit quaternions for kinematic simulation. These static paths are then processed through our simulation to yield dynamically rich, time-indexed trajectories $\tau = \{(t_j, \mathbf{p}_j, \mathbf{q}_j, v_j, w_j)\}_{j=1}^{T}$.

D. Instantiated Rule Vocabulary

We define a core rule vocabulary $\mathcal{R}$ consisting of parametric transformations that encode foundational expert painting skills. Rather than modeling full trajectories, these rules conditionally modify the local execution profile based on the current geometric segment type (e.g., straight paths or corners). Specifically, **Velocity Scaling** (ρ_{vel}) applies a continuous scaling factor to the target linear or angular speed, while **Orientation Offsets** (ρ_{ori}) introduce systematic tilts (represented as a quaternion offset $\boldsymbol{\gamma} \in \mathbb{S}^3$) relative to the surface normal to mimic an expert's angle of attack.

E. Multimodal Rule Inference

To infer the applied rules from demonstrations, we employ a multimodal neural architecture. The model takes a kinematic trajectory and the underlying target CAD model as inputs. To ensure stable neural network training, the trajectory orientations are mapped to a continuous 6D rotation representation. The sequential trajectory data (time, position, 6D orientation) is then processed using a Recurrent Neural Network (RNN) to capture temporal execution dynamics, while the CAD model is sampled into a point cloud and processed via PointNet to extract global geometric features. These two modalities are fused using a cross-attention mechanism, enabling the network to correlate local execution variations with the broader geometric context. Finally, the network utilizes two output heads: a **regression head** that predicts the numerical parameter of the applied rule (e.g., the specific speed multiplier or tilt angle), and a **classification head** that predicts the geometric segment type to which the rule was applied.

F. Transfer to Unseen Geometries

Given the inferred rules and their target segment types, skill transfer is executed on a geometrically planned path τ^* for a novel workpiece. The new path is segmented, classified, and processed through the virtual end-effector simulation. During tracking, the dynamically inferred constraints—such as velocity modifiers and angular offsets—are injected into the corresponding target segments. This yields a refined trajectory $\hat{\tau}$ that preserves the required geometric structure while successfully incorporating the expert's motor strategies.

G. Experimental Results

We assess rule classification and parameter regression across simulated datasets, alongside preliminary insights from real-world human demonstrations.

Simulated Rule Classification: On the foundational L-shape geometries, the multimodal network perfectly correlates the kinematic profile with the rule type, achieving a

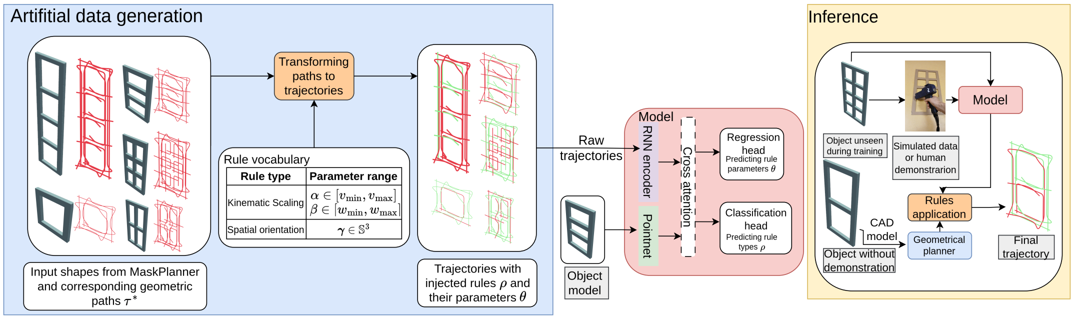

Fig. 1. Overview of the proposed rule-based skill injection framework, illustrating the pipeline from synthetic training data generation to skill transfer on novel geometric paths.

global F1-score of 1.00. On the more complex Window geometries, task difficulty naturally increases. Here, the model achieves F1-scores of 0.64 for velocity scaling and 0.41 for orientation rules.

Simulated Parameter Regression: Continuous parameter extraction is evaluated via Mean Absolute Error (MAE) between predicted and actual values. On L-shapes, the network exhibits highly precise regression with MAEs ranging from 0.02 to 0.09 across both straight and corner segments. On Window topologies, the model maintains strong performance on velocity estimation on straight paths (MAE of 0.10) but estimates all orientations as no rule (MAE 0.30 for orientation). For the complex corners on the windows dataset, the network does not estimate correctly changes in orientation and velocity, most probably due to quicker changes in these values due to the smaller length of the segments (MAE of 0.75 for velocity). Overall, the simulated results confirm the architecture successfully disentangles dynamic overrides from the geometric trajectory.

Preliminary Sim-to-Real Transfer: While the current results validate the architecture on synthetic data, bridging the sim-to-real gap remains a critical next step. We conducted preliminary evaluations on physical human demonstrations recorded via the RoboTwin RoboTeach tool [12]. However, these initial tests revealed a performance drop when transferring from simulation to reality. This gap is primarily attributed to the low sampling rate of the RoboTeach interface, which currently bottlenecks the capture of nuanced, high-frequency human motor skills. Addressing this hardware limitation to enable robust physical skill transfer is a core focus of our ongoing work.

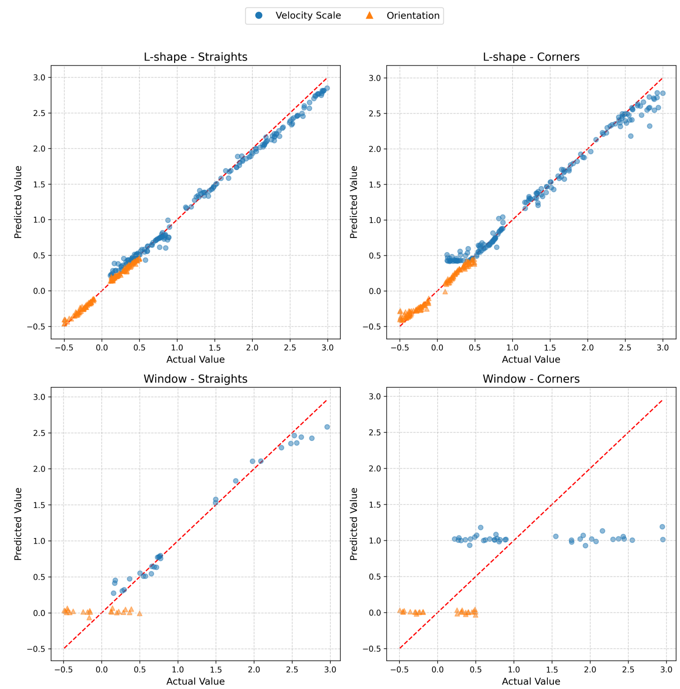

Fig. 2. Predicted vs. Actual values for all combined rules (velocity scaling and orientation offset) evaluated on simulated datasets.

V. DISCUSSION AND FUTURE WORK

In this extended abstract, we introduced a modular framework that extracts and injects execution-level motor expertise into geometrically planned paths. By framing expert behavior as interpretable, atomic rules, such as velocity scaling and orientation offsets, we demonstrated that execution strategies can be successfully decoupled from the underlying geometry in both L-shaped and window topologies. Notably, testing on window frames validates that geometric proximity rules allow the transformer model to infer execution changes based on global CAD features, such as structural crossings, which are not directly observable from the path alone.

While our simulated evaluation validates the extraction of foundational rules, bridging the sim-to-real gap remains a critical next step. As observed in our preliminary tests, capturing nuanced ”painter know-how” relies on high-frequency motor adjustments, and teaching interfaces like the RoboTeach tool often suffer from low sampling rates, presenting a hardware bottleneck. Ongoing work focuses on overcoming these limitations to collect high-fidelity human demonstrations on physical prototypes. Furthermore, we plan to expand the rule vocabulary to include spatial offsets (e.g., controlling spray width via distance to the surface) and strategic periodic patterns. By collecting this high-quality physical data and expanding the atomic rule set, we aim to show that complex, geometry-agnostic expert heuristics can emerge from combinations of simple, interpretable transformations.

References

[1] H. Chen, W. Sheng, N. Xi, M. Song, and Y. Chen, "Automated robot trajectory planning for spray painting of free-form surfaces in automotive manufacturing," vol. 1, pp. 450–455, 2002.

[2] D. Gleeson, S. Jakobsson, R. Salman, F. Ekstedt, N. Sandgren, F. Edelvik, J. S. Carlson, and B. Lennartson, "Generating optimized trajectories for robotic spray painting," *IEEE Transactions on Automation Science and Engineering*, vol. 19, no. 3, pp. 1380–1391, 2022.

[3] M. Lewke, H. Wu, A. List, F. Gärtner, T. Klassen, and A. Fay, "Trajectory optimization for repair by robot-guided cold spray," in *International Thermal Spray Conference*, vol. 84864. ASM International, 2024, pp. 123–129.

[4] N. Ratliff, J. A. (drew) Bagnell, and M. Zinkevich, "Maximum margin planning," in *Proceedings of (ICML) International Conference on Machine Learning*, July 2006, pp. 729 – 736.

[5] S. Ross, G. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, G. Gordon, D. Dunson, and M. Dudík, Eds., vol. 15. Fort Lauderdale, FL, USA: PMLR, 11–13 Apr 2011, pp. 627–635. [Online]. Available: https://proceedings.mlr.press/v15/ross11a.html

[6] W. Montgomery and S. Levine, "Guided policy search via approximate mirror descent," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, ser. NIPS'16. Red Hook, NY, USA: Curran Associates Inc., 2016, p. 4015–4023.

[7] T. Johannink, S. Bahl, A. Nair, J. Luo, A. Kumar, M. Loskyll, J. A. Ojea, E. Solowjow, and S. Levine, "Residual reinforcement learning for robot control," in *2019 international conference on robotics and automation (ICRA)*. IEEE, 2019, pp. 6023–6029.

[8] Y. P. Pane, S. P. Nageshrao, J. Kober, and R. Babuška, "Reinforcement learning based compensation methods for robot manipulators," *Engineering Applications of Artificial Intelligence*, vol. 78, pp. 236–247, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0952197618302446

[9] A. Bucker, L. Figueredo, S. Haddadinl, A. Kapoor, S. Ma, and R. Bonatti, "Reshaping robot trajectories using natural language commands: A study of multi-modal data alignment using transformers," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 978–984.

[10] G. Tiboni, R. Camoriano, and T. Tommasi, "Paintnet: Unstructured multi-path learning from 3d point clouds for robotic spray painting," *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3857–3864, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:257365990

[11] ——, "Maskplanner: Learning-based object-centric motion generation from 3d point clouds," *arXiv preprint arXiv:2502.18745*, 2025.

[12] "RoboTwin: Easy robot teaching - 3 steps & minute," https://robotwin.cz/, Oct. 2022.