arXiv:2605.24881v2 [cs.RO] 29 May 2026

Learning Transferable Motor Skills for Geometry-Aware Robotic Surface Tasks

Miroslav David[1], Karla Stepanova[1], and Robert Babuska[1,2]

***Abstract*— Robotic surface-interaction tasks, such as spray painting or welding, require both accurate geometric planning and precise 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 task 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 orientation offsets, that systematically modify a geometrically planned reference path. We train a multimodal neural network to infer rule parameters jointly from kinematic trajectory data and CAD model geometry. We evaluate our approach through dynamic simulation on L-shaped and window-shaped objects, demonstrating on simulated data that the model successfully extracts velocity and orientation rules across both topologies.**

I. INTRODUCTION

Robotic surface-interaction tasks, such as spray painting or welding, require not only accurate geometric motion planning but also precise execution strategies. While modern motion planners [1], [2], [3] 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 and orientation according to the local geometry of the workpiece, producing execution patterns that are difficult to capture using purely geometric planning methods [4], [5]. An alternative approach is to directly show the desired robot motions through human demonstrations [6]. 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. Expert behavior is represented as a set of interpretable motor rules, such as velocity scaling and orientation offsets, that modify the execution of a geometrically planned path, decoupling planning from execution-level expertise. A multimodal neural network combining a 1D convolutional trajectory encoder and a PointNet++ [7] CAD encoder is used to infer segment types and rule parameters from trajectory data, and the extracted rules are injected into a physics-based simulation to refine geometrically planned trajectories.

The contributions of this work are: (1) A modular framework that represents expert motor skills as a parametric rule vocabulary and injects them into geometrically planned trajectories. (2) A multimodal neural architecture combining a 1D convolutional trajectory encoder and a PointNet++ CAD encoder to infer rule parameters from demonstration data.

II. RELATED WORK

Classical approaches to robotic spray painting primarily focus on geometric coverage and physics-based deposition models [8], [9], [10]. A different line of work addresses skill acquisition through learning from demonstrations and policy learning [11], [12], [13]. 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.

Movement representation frameworks such as Dynamic Movement Primitives (DMPs) offer a compact way to encode and reproduce motion skills through structured dynamical systems [14], [15]. Probabilistic movement primitives extend this idea by representing distributions over trajectories and enabling adaptation and blending [16]. These approaches provide reusable motion representations, but they typically encode behavior at the trajectory level and are less naturally suited to expressing isolated, interpretable execution rules tied to local geometric context.

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 [17], [18], while transformer-based models have been used to reshape existing trajectories from multimodal inputs such as natural language [19]. In robotic spray painting specifically, recent learning-based planners such as PaintNet [20] and MaskPlanner [1] predict geometric path structures directly from 3D object representations. These methods demonstrate strong generalization in geometric path generation but often rely on

Presented at the Workshop on Geometry in the Age of Data-Driven Robotics, ICRA 2026, Vienna, Austria.

[1] Department of Robotics and Machine Perception, Czech Institute of Informatics, Robotics, and Cybernetics, Czech Technical University in Prague, Czech Republic.

[2] Cognitive Robotics, Faculty of 3mE, Delft University of Technology, The Netherlands.

This paper was supported by European Union's Horizon Europe research and innovation programme under grant agreement number 101214398 (ELLIOT). Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or the European Commission. This work was also supported by the Ministry of Education, Youth and Sports of the Czech Republic through the e-INFRA CZ (ID:90254). This research was co-funded by the European Union under the project Robotics and Advanced Industrial Production (reg. no. CZ.02.01.01/00/22_008/0004590).

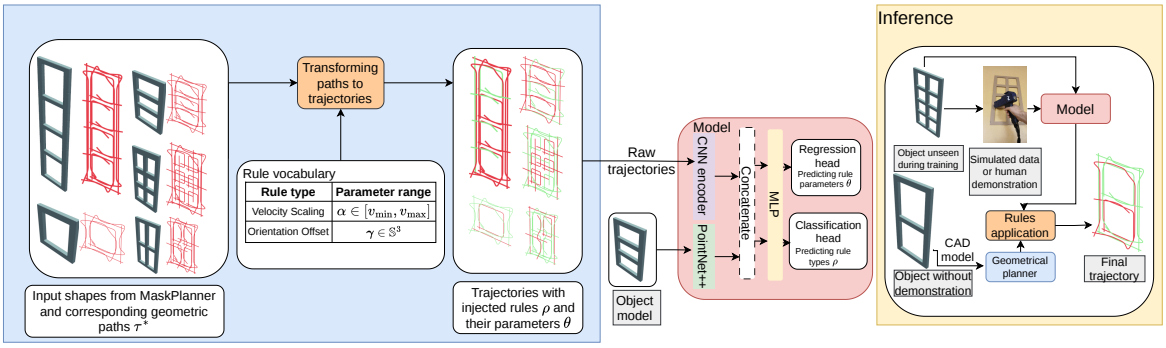

Fig. 1. Overview of the proposed rule-based skill injection framework, illustrating the pipeline from synthetic training data generation to skill transfer on novel geometric paths.

implicit latent representations and do not explicitly separate execution-level expertise from the underlying geometric plan.

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 $\tau_i = \{(t_j, \mathbf{p}_j, \mathbf{q}_j, v_j)\}_{j=1}^{T_i}$ is a time-indexed sequence consisting of positions $\mathbf{p}_j \in \mathbb{R}^3$, orientations as unit quaternions $\mathbf{q}_j \in \mathbb{S}^3$, and linear speed $v_j \in \mathbb{R}$.

Given $\mathcal{D}$, our goals are to: (1) extract a set of interpretable rules $\mathcal{R} = \{\rho_1, \ldots, \rho_m\}$ and associated parameter spaces Θ that characterize expert skills relative to the workpiece geometry, (2) learn an inference function $f : \mathcal{T} \times \mathcal{O} \to \mathcal{C}^m \times \Theta$ that, for each rule ρ_k, predicts a target geometric segment class $c_k \in \mathcal{C} = \{\texttt{straight}, \texttt{corner}, \texttt{none}\}$ and a continuous execution parameter $\theta_k \in \Theta$, and (3) inject the inferred rules into a raw geometric trajectory τ^* for a new workpiece O^*. A rule ρ_k is considered active when $c_k \neq \texttt{none}$. The skill-enhanced trajectory $\hat{\tau}$ is then produced as $\hat{\tau} = (\rho_n(\theta_n, O^*) \circ \cdots \circ \rho_1(\theta_1, O^*))(\tau^*)$, where only active rules are applied to their corresponding segment type.

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).

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 apply systematic modifications to velocity or orientation for each matching segment type.

B. Trajectory Refinement via Dynamic Simulation

To ensure physically continuous dynamics and avoid instantaneous velocity jumps, a pure pursuit controller tracks a lookahead pose along the reference path, applying independent proportional forces and torques to position and orientation errors. The resulting wrench is numerically integrated via fourth-order Runge-Kutta (RK4), modeling the end-effector as a rigid body with mass, inertia, and viscous damping.

C. Data Generation and Preprocessing

Training data are generated using custom L-shaped topologies alongside MaskPlanner window geometries [1] to evaluate generalization across varying surface orientations and structural crossings. Raw geometric paths $(x, y, z, \text{Euler angles}, partID)$ are segmented by part ID, 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 $\tau = \{(t_j, \mathbf{p}_j, \mathbf{q}_j, v_j)\}_{j=1}^{T}$.

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 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 tool tilt.

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 [21]. The resulting 10-dimensional trajectory features, comprising a time delta (1D), a 3D linear velocity vector, and the 6D rotation representation, are processed by a 1D convolutional encoder comprising three stacked Conv1d layers followed by masked average pooling over valid timesteps, producing a compact trajectory embedding. The CAD model is sampled into a point cloud and encoded by a PointNet++ encoder, yielding a global geometric feature vector. The two modality embeddings are combined via concatenation and passed through a shared MLP, enabling the network to jointly reason over local execution dynamics and the broader geometric context. Although the current rule vocabulary does not directly condition on workpiece features, the CAD encoder is included to support future proximity-based rules that require spatial context from the object structure. Finally, the network utilises 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 path is segmented automatically: a sliding window of fixed width is swept along the path, and SVD is applied to each windowed point set to measure local curvature; points whose residual distance from the dominant principal axis falls below a threshold are labelled `straight`, and the remainder `corner`. No manual annotation is required, making the procedure applicable to any new geometry. The inferred rules are then injected into their corresponding segment types during simulation, yielding a refined trajectory $\hat{\tau}$ that preserves the required geometric structure while incorporating the expert's motor strategies.

V. EXPERIMENTAL RESULTS

We assess rule classification and parameter regression across simulated datasets. For simpler L-shaped geometries, 10,000 synthetic trajectories were generated and split 8,000 / 1,000 / 1,000 (train / val / test). For more complex window geometries, the total of 1,000 trajectories were split 800 / 100 / 100 (train/val/test).

Simulated Rule Classification: On the simpler L-shape geometries, the multimodal network achieves a combined F1-score of 0.999 across three classes (straight, corner, none) (see Fig. 2, top). On the more complex window geometries (see Fig. 2, bottom), the model achieves F1-scores of 1.000 for velocity scaling and 0.984 for orientation rules, showing that the model generalises well to more complex geometries.

Simulated Parameter Regression: Continuous parameter extraction is evaluated via Mean Absolute Error (MAE) between predicted and actual values. On L-shapes, the network achieves MAEs of 0.020 rad (orientation offset) and 0.056 (velocity scaling factor, dimensionless, range 0.1–3.0). On window topologies, the model maintains accurate orientation estimation (MAE of 0.038 rad) while velocity scaling shows a moderate error of 0.131 (dimensionless), reflecting the shorter and more rapidly changing segments characteristic of window frame geometries. This suggests that finer-grained segmentation or segment-length normalisation may be needed to handle geometries with dense curvature changes. Overall, the simulated results confirm that the architecture successfully disentangles dynamic overrides from the geometric trajectory.

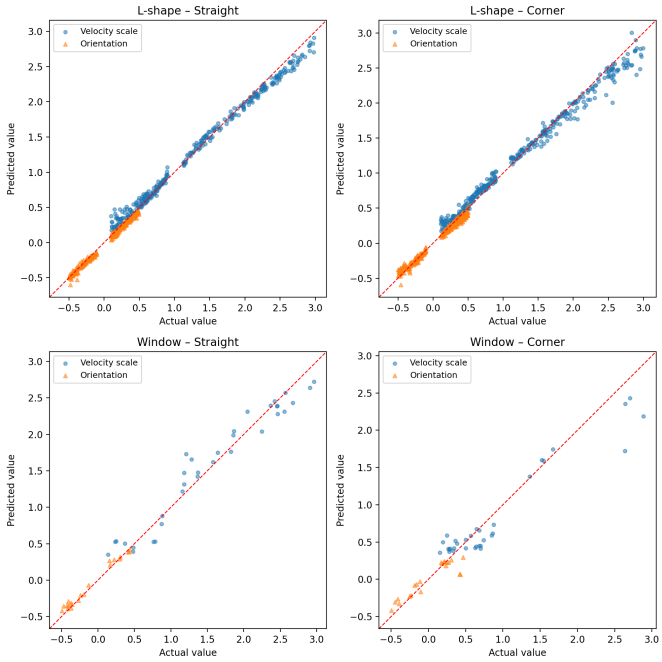

Fig. 2. Predicted vs. actual parameter values on the test set. Each subplot shows one geometry type: L-shape straight (top-left), L-shape corner (top-right), window straight (bottom-left), window corner (bottom-right). Blue circles: velocity scaling factor; orange triangles: orientation offset (rad). The red dashed line indicates perfect prediction.

VI. DISCUSSION AND FUTURE WORK

We demonstrated that execution strategies can be successfully decoupled from workpiece geometry using interpretable, atomic rules in both L-shaped and window topologies. The current rule vocabulary operates purely on segment type and does not yet exploit workpiece geometry at inference time; rules conditioned on runtime proximity to CAD features such as edges or crossings would require the network to perform structured spatial reasoning, which motivates the inclusion of the CAD encoder in anticipation of this extension. Bridging the sim-to-real gap remains the critical next step: physical teaching interfaces currently suffer from low sampling rates, limiting capture of high-frequency motor adjustments. Future work will address this bottleneck, expand the rule vocabulary to include proximity-based rules conditioned on structural CAD features such as crossings or edges, and evaluate zero-shot transfer to held-out topologies such as U-shaped and T-shaped geometries.

REFERENCES

[1] G. Tiboni, R. Camoriano, and T. Tommasi, "Maskplanner: Learning-based object-centric motion generation from 3d point clouds," *arXiv preprint arXiv:2502.18745*, 2025.

[2] P. Rabino, G. Tiboni, and T. Tommasi, "Foldpath: End-to-end object-centric motion generation via modulated implicit paths," 2025. [Online]. Available: https://arxiv.org/abs/2511.01407

[3] W. Sheng, N. Xi, M. Song, and Y. Chen, "Cad-based automated robot trajectory planning for spray painting of free-form surfaces," *Industrial Robot: An International Journal*, vol. 29, pp. 426–433, 10 2002.

[4] S. Nieto Bastida and C.-Y. Lin, "Autonomous trajectory planning for spray painting on complex surfaces based on a point cloud model," *Sensors*, vol. 23, no. 24, 2023. [Online]. Available: https://www.mdpi.com/1424-8220/23/24/9634

[5] R.-X. Hua, H.-X. Ma, W. Zou, W. Zhang, and Z. Wang, "A spraying path planning algorithm based on point cloud segmentation and trajectory sequence optimization," *Int. J. Control Autom. Syst.*, vol. 22, no. 2, pp. 615–630, Feb. 2024.

[6] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn, "Learning fine-grained bimanual manipulation with low-cost hardware," 2023. [Online]. Available: https://arxiv.org/abs/2304.13705

[7] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *Advances in neural information processing systems*, vol. 30, 2017.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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

[13] 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.

[14] A. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal, "Dynamical movement primitives: Learning attractor models for motor behaviors," *Neural Computation*, vol. 25, pp. 328–373, 02 2013.

[15] M. Saveriano, F. Abu-Dakka, A. Kramberger, and L. Peternel, "Dynamic movement primitives in robotics: A tutorial survey," *The International Journal of Robotics Research*, vol. 42, 09 2023.

[16] A. Paraschos, C. Daniel, J. R. Peters, and G. Neumann, "Probabilistic movement primitives," in *Advances in Neural Information Processing Systems*, C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, Eds., vol. 26. Curran Associates, Inc., 2013. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2013/file/e53a0a2978c28872a4505bdb51db06dc-Paper.pdf

[17] 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.

[18] 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

[19] 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.

[20] 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

[21] Y. Zhou, C. Barnes, J. Lu, J. Yang, and H. Li, "On the continuity of rotation representations in neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5745–5753.