RFMGrasp: Riemannian Flow Matching for Grasp Generation

Ali Batur Karakullukcu
Technical University of Munich (TUM)
ali.karakullukcu@tum.de

Ferdy Dermawan Hadiwijaya
Technical University of Munich (TUM)
ferdydh.hadiwijaya@tum.de

***Abstract*—Robot grasp generation requires finding suitable SE(3) poses that combine position and orientation. While recent diffusion-based approaches have shown promise, they face computational overhead and often ignore the distinct geometric nature of translations and rotations. We present RFMGrasp, which uses Riemannian flow matching to generate grasp poses by separately modeling translations in $\mathbf{R}^3$ and rotations in SO(3). Our key innovation lies in a specialized SO(3) loss with skew projection that maintains valid rotations during flow matching. By directly learning optimal transport paths while respecting rotation geometry, our method avoids the overhead of diffusion models. Trained on the Acronym dataset, RFMGrasp generates high-quality grasps for diverse objects, including those not seen during training, demonstrating that proper geometric handling of rotations enables efficient and generalizable grasp generation.**

I. Introduction

To manipulate objects effectively, robots must first grasp them reliably. While researchers have studied robotic grasping for decades, robots still struggle to grasp objects as flexibly as humans do [1]. For parallel-jaw grippers like the Franka Panda, a key challenge lies in finding grasps that are both stable enough to maintain contact during manipulation and robust to real-world uncertainties in perception and control. These grasps must specify both where to position the gripper and how to orient it relative to the target object.

Recent approaches have turned to generative models [2]–[5], using large-scale grasp datasets to learn how to generate and represent grasps. These methods have succeeded in capturing the data distribution of successful grasps. Particularly, Urain et al. (2023) [6] developed a diffusion model constrained to Riemannian manifolds, which enables representation of grasp configurations that respect their geometric structure.

However, diffusion models rely on learning noise-to-signal transitions through multiple forward-backward processes, creating computational overhead. While flow matching has emerged as "a more robust and stable alternative for training diffusion models" [7], current approaches like FFHFlow [8] focus on learning joint configurations and 6D poses in Euclidean space. This limits their ability to capture the true geometric structure of grasp poses, which live in the special Euclidean group SE(3).

We address these limitations with RFMGrasp, a novel approach combining the stability of flow matching with explicit modeling of SE(3)'s geometric structure. A grasp pose in SE(3) consists of two fundamentally different components: a translation in R^3 specifying the gripper's position, and a rotation in SO(3) defining its orientation. Unlike translations, which can be freely added and scaled, rotations must preserve specific geometric properties like orthogonality. This inherent structure demands specialized treatment that existing flow matching approaches don't provide.

Our method stands apart by treating translations and rotations according to their natural mathematical properties. For translations, we employ standard Euclidean flows in R^3. For rotations, we introduce a novel loss function using skew projection to ensure generated rotations remain valid elements of SO(3) throughout the flow matching process. This specialized handling maintains geometric constraints while allowing smooth optimization. By directly learning optimal transport paths between distributions, our flow matching approach provides more stable training and generation compared to diffusion's forward-backward processes.

We trained RFMGrasp on the ACRONYM dataset [9] and found that it generates plausible grasps across diverse objects.

In conclusion, we propose a novel grasp generation framework that applies flow matching directly in SE(3), ensuring stability by treating translations and rotations separately. To maintain geometric validity, we introduce a specialized SO(3) loss with skew projection. Our method generalizes well to unseen objects, achieving a 98.77% grasp success rate when generating multiple candidates.

II. Related Work

Generative Models for Grasp Generation. Recent work has shown promising results using generative approaches instead of reinforcement learning for grasp synthesis. Feng et al. (2024) [8] introduced Conditional Flow Matching (CFM) through their FFHFlow model, focusing on generating 21-dimensional robot joint configurations. While FFHFlow demonstrates strong performance, its Euclidean latent space structure limits its ability to handle the geometric constraints of object orientations needed for precise parallel-jaw grasps.

Learning in SE(3) Space. Modeling grasps directly in the 6-dimensional SE(3) space offers advantages over joint-space representations. Flow matching in SE(3) can learn distributions of successful grasps in a reduced dimensional space, which improves data efficiency compared to learning in the full joint

space while preserving the physical correctness of rigid body transformations.

Generative Models with Geometric Constraints. De et al. (2022) [10] showed that incorporating Riemannian geometry into score-based generative models improves performance on data that exists on manifolds, from robotics to protein modeling. Braun et al. (2024) [11] developed the Riemannian Flow Matching Policy for robot motion learning, which maintains SO(3) manifold structure during generation. Our approach builds on work by Urain et al. (2023) [6], who combined diffusion models with Riemannian geometry for SE(3) generation. While Bose et al. (2023) [12] established the mathematical framework for SE(3) stochastic flow matching in protein generation, we apply this technique specifically to grasp synthesis by conditioning the model on object signed distance fields (SDFs).

Our method applies these geometric principles to grasp generation by treating translations and rotations distinctly and preserving SO(3) constraints through skew projection in velocity computations. This approach generates grasps across diverse object geometries while maintaining consistency in the SE(3) space.

III. Methods

A. Dataset

We use the Acronym dataset [13], containing 17.7 million simulated grasps generated by a parallel-jaw gripper for objects like bottles and furniture. Each object mesh is converted to a watertight format and sampled to create a Signed Distance Function (SDF), with a grid structure around the mesh. Each entry specifies the gripper's action space (position and orientation) as a 4x4 transformation matrix, with the environment defined by the 3D object shape in SDF format. For the training, only successful grasps are used.

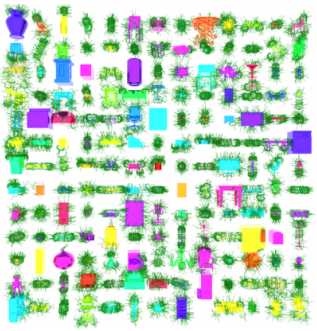

Fig. 1. Acronym dataset which has 8872 objects with 8.8 million successful grasps.

B. Preliminaries

1) Projection onto the Lie Algebra: The special Euclidean group SE(3), which describes 3D rigid transformations, can be considered as a product of SO(3) and $\mathbb{R}^3$. This means that each element of SE(3) can be decomposed into a rotational component (an element of SO(3)) and a translational component (an element of $\mathbb{R}^3$).

SE(3) is a globally curved manifold M. Manifolds can be approximated with flat (Euclidean) space T_xM in an infinitesimally small neighborhood around a point x on it. More formally, at each point $x \in M$, the tangent space T_xM is a real vector space that is isomorphic to $\mathbb{R}^n$, where n is the dimension of the manifold M.

2) Riemannian Flow Matching:

a) Probability paths in SE(3): A probability path $\{\rho_t\}_{t\in[0,1]}$ is a continuous family of probability distributions that connects two given distributions ρ_0 and ρ_1.

A flow ψ_t on M is a one-parameter diffeomorphism defined by the ODE:

$$\frac{d}{dt}\,\psi_t(x) \;=\; v_t\big(\psi_t(x)\big), \quad \psi_0(x) \;=\; x. \tag{1}$$

There are infinitely many ways to create a velocity field to reach our target distribution. In practice, constant velocity vector fields are often used in training [12].

b) Flow Matching (FM).: Consider a path of distributions $\{\rho_t\}_{t\in[0,1]}$ connecting $\rho_0 = \rho_{\text{rand}}$ and $\rho_1 = \rho_{\text{data}}$ via a velocity field $\{u_t\}_{t\in[0,1]}$. We can learn a continuous normalizing flow by regressing a parametric vector field v_θ to match u_t [7]. The standard Flow Matching (FM) objective is

$$\mathcal{L}_{\text{FM}}(\theta) \;=\; \mathbb{E}_{t,\,x\sim\rho_t}\big\|v_\theta(t,x) \;-\; u_t(x)\big\|_g^2, \tag{2}$$

where $\|\cdot\|_g$ denotes the chosen distance metric.

c) Conditional Flow Matching (CFM).: Since u_t is unavailable in closed form, we learn a *conditional* velocity field $u_t(x \mid z)$, creating a path from a random point to any selected data point. Even then, the unconditional path remains $\rho_t(x) = \int \rho_t(x \mid z)\, q(z)\, dz$. Marginalizing $u_t(\cdot \mid z)$ recovers the original velocity field:

$$u_t(x) \;=\; \int u_t(x \mid z)\, \frac{\rho_t(x \mid z)\, q(z)}{\rho_t(x)}\, dz. \tag{3}$$

We then regress v_θ against $u_t(x \mid z)$, yielding the CFM objective [14]:

$$\mathcal{L}_{\text{cfm}} \;=\; \mathbb{E}_{t\sim U(0,1),\, z\sim q(z),\, x\sim\rho_t(\cdot|z)}\Big\|v_\theta(t,x) - u_t(x \mid z)\Big\|_g^2. \tag{4}$$

Since $\nabla_\theta \mathcal{L}_{\text{cfm}}(\theta)$ coincides with $\nabla_\theta \mathcal{L}_{\text{FM}}(\theta)$, CFM retains the same solution as FM but is tractable even when $u_t(x)$ is not known explicitly.

d) Sampling Procedure for Training: To generate a conditioning path, we first sample a random grasp configuration $(R_{\text{rand}},\, t_{\text{rand}})$ and a data grasp configuration $(R_{\text{data}},\, t_{\text{data}})$. from our dataset. We then draw a random time parameter t uniformly from the interval $[0, 1]$.

For the translation component, we interpolate linearly as:

$$t(t) \;=\; t_{\text{rand}} \;+\; \big(t_{\text{data}} - t_{\text{rand}}\big)\, t \tag{5}$$

$$\dot{t} \;=\; \frac{t_{\text{data}} \;-\; t_{\text{rand}}}{t + \epsilon}. \tag{6}$$

where ϵ is a small constant to avoid division by zero.

For the rotation component, we need to use exponential and logarithmic maps to interpolate along the geodesic on $SO(3)$ between R_{rand} and R_{data}.

The exp map written from the identity rotation can be calculated as the standard matrix exponential. In addition, we use the method of [12] where log map converts R into its axis-angle (rotation vector) representation, and rotation vector into a skew-symmetric matrix in $\mathfrak{so}(3)$. This avoids the costly infinite series computation of the matrix logarithm.

Having said this instantaneous rotation $R(\mathrm{t})$ can be calculated as below:

$$R(\mathrm{t}) \;=\; R_{\text{rand}}\,\exp\Big(\mathrm{t}\,\log\big(R_{\text{rand}}^{\top} R_{\text{data}}\big)\Big). \tag{7}$$

Consequently, the instantaneous rotation velocity $\dot{R}(\mathrm{t})$ can be written as:

$$\dot{R}(\mathrm{t}) \;=\; R(\mathrm{t})\,\log\Big(R_{\text{rand}}^{\top}\,R(\mathrm{t})\Big)\;\Big/\;\big(\mathrm{t}+\epsilon\big), \tag{8}$$

Hence, at each time $\mathrm{t}\in[0,1]$, we obtain a pose $\big(R(\mathrm{t}),\,t\big)$ on SE(3).

Considering that each object has a different grasp distribution, we should also condition on the geometry of the target object $\bar{m}$, therefore we end up with the target loss function:

$$L_{SO(3)}(\mathrm{t}) = \Big\| v_\theta\big(\mathrm{t},\mathrm{T_t},\bar{m}\big) \;-\; \frac{\log R_{\mathrm{t}}\big(R_{\text{data}}\big)}{\mathrm{t}}\Big\|^2_{SO(3)}. \tag{9}$$

$$L_{\mathbb{R}^3}(\mathrm{t}) = \Big\| v_\theta\big(\mathrm{t},\mathrm{T_t},\bar{m}\big) \;-\; \frac{t_{\text{data}}-t_{\text{rand}}}{\mathrm{t}}\Big\|^2_{2}. \tag{10}$$

$$\mathcal{L} = \mathbb{E}_{\mathrm{t}\sim\mathcal{U}(0,1)}\Big[L_{SO(3)}(\mathrm{t}) \;+\; L_{\mathbb{R}^3}(\mathrm{t})\Big]. \tag{11}$$

3) Network Architecture and SDF Encoding: For the grasp generation task, our network incorporates a CNN-based SDF encoder to extract geometric feature encodings from the object's signed distance field (SDF). Each SDF is first normalized to the range $[-1,1]$ on a per-instance basis, and a scaling factor is computed and passed along with the encoded features. The normalized SDF encoding is concatenated with the gripper's current pose information, and this composite feature vector is fed into an MLP that predicts the flow vector field.

Once the MLP produces the velocity field, we split the output into its translation and rotation components. The translation component is a standard vector in $\mathbb{R}^3$. However, for the rotation part, we need to ensure that the velocity lies in the tangent space of SO(3). In practice, we treat the MLP's rotational output as a 3×3 matrix ξ, and then project it onto the space of skew-symmetric matrices (the Lie algebra $\mathfrak{so}(3)$). Concretely, we compute:

$$\omega \;=\; \frac{1}{2}\,\big(\xi \;-\; \xi^{T}\big), \tag{12}$$

where ω is guaranteed to be skew-symmetric. Geometrically, ω represents the instantaneous angular velocity on SO(3). Then, the change in rotation matrix, $\dot{R}$ is obtained by

$$\dot{R} \;=\; R\,\omega, \tag{13}$$

so that the rotation $R\in\mathrm{SO}(3)$ remains on the manifold.

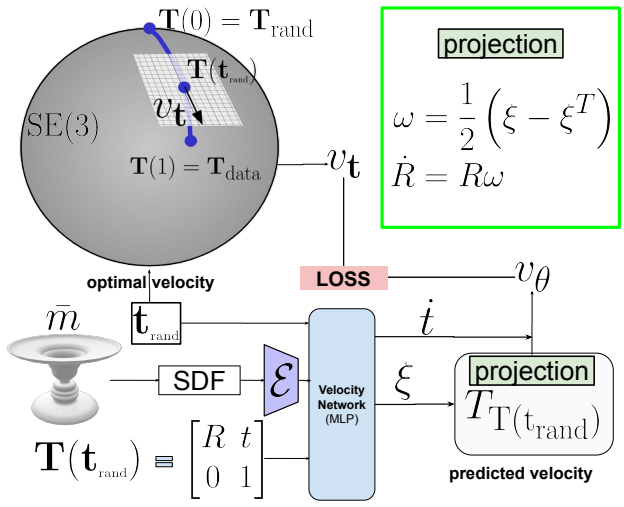

Fig. 2. Architecture overview. Network outputs translational velocity and a rotational velocity which is projected to the tangent space at the current transformation.

4) Memory Limitations: Storing full 3D grid-based SDFs can be memory-intensive when using large batch sizes, since each sample would require $(B\times D\times D\times D)$ space. To address this, we generate multiple trajectories (K) for each SDF after a single forward pass through the encoder, effectively reducing the memory footprint to $\big(\frac{B}{K}\times D\times D\times D\big)$. The latent embeddings are then duplicated to reconstruct an effective batch size of B. This approach enables us to handle larger batches without exceeding memory limits.

5) Inference Procedure: During inference, our goal is to generate new grasp configurations by integrating the learned velocity field from $\mathrm{t}=0$ to $\mathrm{t}=1$. Let $\big(R(0),\,t(0)\big)$ be the random initial grasp pose and $\Delta\mathrm{t}=1/N$ for some integer N for discretization. At each discrete time step $\mathrm{t}_i = i\,\Delta\mathrm{t}$, we evaluate the velocity network:

$$\big(\dot{R}(\mathrm{t}_i),\;\dot{t}(\mathrm{t}_i)\big) \;=\; v_\theta\Big(\mathrm{t}_i,\;\big(R(\mathrm{t}_i),\,t(\mathrm{t}_i)\big),\;\bar{m}\Big), \tag{14}$$

The translation is updated via an Euler step:

$$t\big(\mathrm{t}_{i+1}\big) \;=\; t\big(\mathrm{t}_i\big) \;+\; \Delta\mathrm{t}\,\dot{t}(\mathrm{t}_i). \tag{15}$$

Since $\dot{R}(\mathrm{t}_i)\in\mathfrak{so}(3)$ is represented internally as a skew-symmetric matrix (angular velocity), we use the matrix exponential to stay on the manifold $SO(3)$. Concretely,

$$R\big(\mathrm{t}_{i+1}\big) \;=\; R\big(\mathrm{t}_i\big)\,\exp\Big(\dot{R}(\mathrm{t}_i)\,\Delta\mathrm{t}\Big). \tag{16}$$

This exponential map ensures $R(\mathrm{t}_{i+1})\in SO(3)$ remains a valid rotation. By repeating these updates for $i=0,1,\ldots,N-1$, we obtain a final grasp pose.

IV. Experimental Results

The goal of a grasp generation model is to produce grasps that are effective in real-world scenarios. Simulation environments serve as an intermediate step before real-life testing.

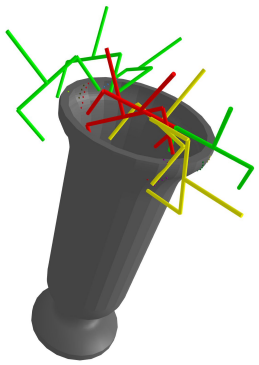

Fig. 3. Visualization of eight generated grasps. Green: Optimal grasps with ideal positioning and no collision. Yellow: Grasps with minor collisions that may still succeed. Red: Gripper unable to reach the object.

However, replicating the exact conditions of the ACRONYM dataset requires access to its original simulation parameters and significant computational resources. Due to these constraints, we deemed this beyond the scope of our project. Instead, our evaluation is based on the representation of objects and the parallel-jaw gripper in Trimesh [15], offering a more practical and computationally efficient testing method that better suits our project's needs.

Our 3D spatial evaluation classifies each generated grasp into one of three categories:

- **Optimal**: The gripper is properly aligned with the object, centered between the jaws, and free of collisions.
- **Collision**: The gripper intersects with the object's geometry. While not necessarily a failure, as some ACRONYM dataset samples are marked successful despite minor collisions, this may affect grasp stability.
- **Out of Reach**: The gripper's position prevents it from making contact with the object.

TABLE I
RFMGRASP PERFORMANCE METRICS

Metric	**Seen Objects**	**Unseen Objects**
Graspable	97.08%	94.24%
Collision-Free	85.74%	76.98%
Optimal	82.85%	71.31%

Graspable: Not "Out of Reach". Collision-Free: Not "Collision". Optimal: Neither "Out of Reach" nor "Collision".

As shown in Table I, the model successfully positions the gripper near objects in 97.08% of cases for seen objects and 94.24% for unseen objects (graspable metric). The optimal rate—indicating both correct positioning and a collision-free grasp—reaches 82.85% for seen objects and 71.31

A key advantage of our generative model is its high throughput. By generating multiple grasp candidates per object rather than relying on a single attempt, we can significantly improve success rates. Testing shows that producing 10 grasp candidates per object boosts the success rate to 99.19% for seen objects and 98.77% for unseen ones. This approach is highly practical for real-world robotic applications, as generating additional grasps incurs minimal computational overhead.

Furthermore, the model's strong performance on unseen objects (98.77% success with 10 candidates) suggests it learns meaningful geometric relationships between objects and grasps rather than merely memorizing specific examples. Notably, this generalization occurs even without extensive hyperparameter tuning, indicating potential for further optimization and improved performance.

A. Distribution Learning Quality

We compared the samples generated by our network to the training distribution for each object using the Wasserstein-1 distance, and examined the impact of various batch sizes on the learned distribution quality. Empirically, larger batch sizes led to better scores.

TABLE II
AVERAGE $\mathcal{W}_1$ DISTANCES FOR DIFFERENT BATCH SIZES.

Batch Size	SO(3)	$\mathbb{R}^3$
128	0.444	0.334
1024	**0.412**	**0.334**

V. FUTURE DIRECTIONS

RFMGrasp has demonstrated strong baseline performance, and there are several potential avenues for further improvement. A systematic hyperparameter optimization could enhance performance, as the current results were obtained without parameter tuning or early stopping criteria.

Moreover, the literature highlights several flow matching variants, such as optimal transport [16] and stochastic flow matching [12], which could offer advantages over our current approach. Exploring these alternatives may lead to further refinements.

Additionally, evaluating our model in a physical simulation environment like PyBullet is crucial, as it would provide deeper insights into its real-world applicability. While geometric metrics serve as an initial validation, simulation can reveal dynamic properties such as force closure and slip resistance. Ultimately, real-world testing would serve as the definitive validation of our proposed method.

VI. CONCLUSION

This paper introduced RFMGrasp, a novel grasp generation method that integrates flow matching with SE(3)-based learning. By respecting the geometric structure of grasp poses, the method appropriately handles translations and rotations according to their intrinsic properties. Experimental results demonstrate strong generalization to novel objects, achieving a 98.77% success rate with 10-grasps candidates setup.

Furthermore, the method's efficiency makes it particularly well-suited for real-world robotic applications with limited computational resources.

REFERENCES

[1] H. Zhang, J. Tang, S. Sun, and X. Lan, "Robotic grasping from classical to modern: A survey," *arXiv preprint arXiv:2202.03631*, 2022.

[2] J. Lundell, F. Verdoja, T. N. Le, A. Mousavian, D. Fox, and V. Kyrki, "Constrained generative sampling of 6-dof grasps," *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2940–2946, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:257050290

[3] Z. Weng, H. Lu, J. Lundell, and D. Kragic, "Gonet: An approach-constrained generative grasp sampling network," *2023 IEEE-RAS 22nd International Conference on Humanoid Robots (Humanoids)*, pp. 1–7, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:257504928

[4] T. R. Player, D. Chang, F. Li, and G. A. Hollinger, "Real-time generative grasping with spatio-temporal sparse convolution," *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7981–7987, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:259325884

[5] G. Chou, Y. Bahat, and F. Heide, "Diffusion-sdf: Conditional generative modeling of signed distance functions," *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2262–2272, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:254017862

[6] J. Urain, N. Funk, J. Peters, and G. Chalvatzaki, "Se (3)-diffusionfields: Learning smooth cost functions for joint grasp and motion optimization through diffusion," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 5923–5930.

[7] Y. Lipman, R. T. Q. Chen, H. Ben-Hamu, M. Nickel, and M. Le, "Flow matching for generative modeling," *October 2022*, 2022.

[8] Q. Feng, J. Feng, Z. Chen, R. Triebel, and A. Knoll, "FFHFlow: A flow-based variational approach for multi-fingered grasp synthesis in real time," *arXiv preprint arXiv:2407.15161*, 2024.

[9] C. Eppner, A. Mousavian, and D. Fox, "ACRONYM: A large-scale grasp dataset based on simulation," in *Under Review at ICRA 2021*, 2020.

[10] V. De Bortoli, E. Mathieu, M. Hutchinson, J. Thornton, Y. W. Teh, and A. Doucet, "Riemannian score-based generative modelling," *Advances in neural information processing systems*, vol. 35, pp. 2406–2422, 2022.

[11] M. Braun, N. Jaquier, L. D. Rozo, and T. Asfour, "Riemannian flow matching policy for robot motion learning," *ArXiv*, vol. abs/2403.10672, 2024. [Online]. Available: https://api.semanticscholar.org/CorpusID:268512755

[12] A. J. Bose, T. Akhound-Sadegh, G. Huguet, K. Fatras, J. Rector-Brooks, C.-H. Liu, A. C. Nica, M. Korablyov, M. Bronstein, and A. Tong, "Se (3)-stochastic flow matching for protein backbone generation," *arXiv preprint arXiv:2310.02391*, 2023.

[13] C. Eppner, A. Mousavian, and D. Fox, "Acronym: A large-scale grasp dataset based on simulation," in *IEEE International Conference on Robotics and Automation*, 2021.

[14] R. T. Q. Chen and Y. Lipman, "Flow matching on general geometries," *arXiv preprint arXiv:2302.03660*, 2024. [Online]. Available: https://arxiv.org/abs/2302.03660

[15] "Trimesh [computer software]," Retrieved from https://github.com/mikedh/trimesh, 2019.

[16] A. Tong, K. Fatras, N. Malkin, G. Huguet, Y. Zhang, J. Rector-Brooks, G. Wolf, and Y. Bengio, "Improving and generalizing flow-based generative models with minibatch optimal transport," *arXiv preprint arXiv:2302.00482*, 2023.