# FLASH: Flow-Based Language-Annotated Grasp Synthesis for Dexterous Hands

**Hrishit Leen*** **Jeremy A. Collins** **Kunal Aneja** **Nhi Nguyen**

**Priyadarshini Tamilselvan** **Sri Siddarth Chakaravarthy P**

**Animesh Garg**

Georgia Institute of Technology

**Abstract:** We introduce FLASH, a method for language-conditioned dexterous grasping that jointly models task intent and physical contact quality for robot hands. Unlike prior approaches, our text-conditioned grasp synthesis pipeline is explicitly aware of geometric information during generation. FLASH learns a single flow-matching model conditioned on hand and object point clouds and natural language instructions. Our model operates on live-updated, vectorized hand meshes and is trained on our improved grasp dataset, FLASH-Drive, which includes refined grasps, water-tight meshes and augmented text annotations. This enables FLASH to outperform prior work in producing physically plausible grasps that align with goals specified via text. We use a pre-trained large language model as the backbone of our architecture, enabling generalization to novel prompts and objects. To the best of our knowledge, we will be the first to release code for our text-conditioned robot grasping pipelines, the weights to our pre-trained models, and a large-scale dataset of text-conditioned dexterous grasps.

**Keywords:** Dexterous Grasping, Flow Matching, Large Language Models

## 1 Introduction

Dexterous robotic grasping has recently made rapid progress, driven by differentiable simulation, large-scale datasets, and pre-trained generative models [1, 2, 3, 4]. However, existing pipelines still separate *physical plausibility* from *task intent*. Geometry-centric methods optimize contact quality while ignoring high-level semantics [5, 6], whereas language-conditioned approaches first sample grasps and then refine or rank them in a second stage [7, 8].

We argue that unifying these objectives requires a model that continuously couples language, geometry, and contact dynamics throughout training. To that end, we introduce FLASH, a conditional flow-matching network that generates stable, semantically aligned grasps for multi-fingered hands.

A key challenge in dexterous robot grasping is creating a large-scale dataset with varying objects, a wide range of grasp types, and corresponding language annotations. In recent years, there has been a shift towards differentiable simulation for generating dexterous grasps, leading to the development of datasets such as Grasp'D, Fast Grasp'D, MultiDex, and DexGraspNet [9, 3, 10, 2]. Another approach involves utilizing hand-object interaction datasets–either by learning grasping generation models directly on human grasps parametrized by the MANO hand model, or by retargeting these human grasps to robot morphologies, such as the Shadow hand. Recent works such as SemGrasp [11] and MultiGraspLLM [4] have used GPT-4o and GPT-4v [12] to create natural language annotations based on finger-object contact information and/or images of real or rendered grasps.

*Corresponding author: hleen3@gatech.edu

9th Conference on Robot Learning (CoRL 2025), Seoul, Korea.

To take advantage of these datasets, grasp generation models have incorporated recent advancements in generative modeling architectures, including autoencoders, transformers, and diffusion models. Autoencoders [13, 5, 14] have been used to represent compress high-dimensional data, such as contact maps and hand-object embeddings, to a structured latent space to enable more efficient learning. Diffusion models [15, 16, 17, 8] have frequently been used to generate diverse sets of grasps while maintaining quality. Transformers [18] have been used for incorporating multimodal data, especially for language-conditioned grasping. These models take advantage of large LLMs to leverage their "common-sense" reasoning and better correlate text conditioning with ground-truth grasp.

To build on these developments, we decide to move forward with using flow-matching as our model backbone architecture. Unlike diffusion-based methods that require multiple sampling steps, flow-matching offers a more straightforward approach and faster inference [19]. Recent models [20, 21] have shown the effectiveness of flow-matching in balancing grasp diversity and quality maintenance.

We develop a flow-matching-based grasp synthesis framework for dexterous robotic hands. Our method maps language prompts and point clouds from the object hand directly to stable, semantically meaningful grasps across various categories of objects and multi-fingered hands. While prior approaches rely on fixed mesh inputs, we introduce GPU-accelerated, vectorized meshes that are regenerated for every model inference call. Additionally, we leverage this same mesh processing infrastructure to post-process existing datasets, enhancing grasp quality and physical feasibility while preserving language adherence. To promote semantic generalization, we condition our model on Qwen-2.5 [22], a 0.5B parameter open-weight LLM, introducing broad world knowledge to improve grounding for unseen prompts and novel object geometries.

In summary, this paper makes the following main contributions:

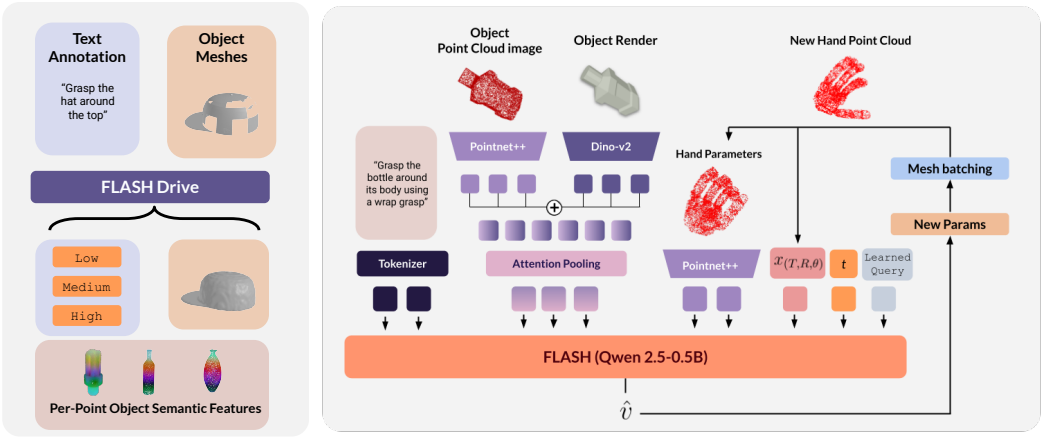
1. FLASH, a state-of-the-art flow-matching architecture that produces contact-level grasp quality for dexterous robotic hands while adhering to language commands.
2. FLASH-drive, a large-scale, language-annotated, high-fidelity robot grasping dataset along with a semantically featured point cloud.
3. An efficient mesh processing method that allows us to not only improve existing language-annotated grasping datasets, but also enables our architecture to be geometrically aware during the flow-matching process.

## 2 Method

Our work introduces FLASH, a novel approach for generating dexterous robotic grasps conditioned on natural language instructions. FLASH employs a conditional flow-matching model trained on FLASH-Drive, our enhanced dataset featuring refined grasps and augmented annotations. This section details the dataset refinement process that yields FLASH-Drive, the architecture and training of the flow-based grasp generation model, and the inference procedure.

### 2.1 Dataset Refinement and FLASH-Drive

Generating high-quality, language-conditioned grasps requires training data that is both physically plausible and semantically rich. We developed FLASH-Drive by significantly improving the Multi-GraspLLM dataset [4]. To improve semantic richness, we employed the state-of-the-art OpenAI o4-mini vision language model, utilizing its function calling capabilities with structured JSON schemas to generate detailed textual descriptions of contact patterns for each grasp given a render of the grasp and the existing low-level annotations of contact per-finger. This process results in over $200\,000$ new text annotations that capture functional intent with nuance. Concurrently, to enhance physical plausibility, we addressed mesh quality issues inherent in some existing datasets that impede accurate collision checking and SDF-based optimization. We implemented a robust pipeline to create watertight object meshes by first voxelizing the original mesh, then using multi-view orthographic depth maps to fill holes, and finally applying Laplace smoothing to ensure surface smoothness suitable for accurate Signed Distance Function (SDF) computation at resolutions of approximately $100^3$. With these high-fidelity meshes and their corresponding SDFs, we refined the original grasp poses. This




**(a)** Dataset **(b)** Architecture

**Figure 1:** FLASH is a conditional flow-matching model capable of generating semantically-aligned grasps that are physically plausible. We first produce a dataset by improving the quality of object meshes, improving the physical feasibility of grasps via SDF-based optimization, adding synthetic text annotations, and generating per-point semantic object features using Dinov2. We then train FLASH on this data with a flow-matching objective, incorporating a live updating hand point cloud for geometric awareness and per-point object features to enable semantic generalization.

refinement utilized efficient batch meshing utilizing kernels from both Pytorch3D and Kaolin (see Sec. 2.2) to optimize grasps directly against the SDF: sampling a point cloud ($N_{refine} = 8192$ points) from the hand mesh and minimizing a loss that penalizes both object penetration ($\text{SDF}_O(p) < 0$) and excessive distance from the surface ($\text{SDF}_O(p) > \epsilon_{dist}$). We also ensure that during the optimization process self-penetration doesn't occur by sampling point-clouds at the fingertips and penalizing the distances between them. We are able to optimize over batches of grasps using this technique, as the hand parameters are optimized independently. The outputs of this optimization process can be visualized in Figure 1a. This process yielded **FLASH-Drive**, a large-scale dataset characterized by physically refined grasps, watertight object meshes, and diverse language annotations, forming the foundation for training FLASH.

### 2.2 Flow-Based Grasp Generation

FLASH learns a single-stage, end-to-end generative process for grasp synthesis, mapping conditioning information directly to stable, semantically relevant grasps. It achieves this using a conditional flow-matching (CFM) framework [21]. The core task is to learn a velocity field $\hat{v}(x_t, t, c)$ that guides the evolution of hand parameters $x_t \in \mathbb{R}^{25} = [T, R, \theta]$, where $T \in \mathbb{R}^3$ is the wrist position, $R \in \mathbb{R}^6$ is the wrist rotation, and $\theta \in \mathbb{R}^{16}$ represents the joint angles of each finger link in radians. $x$ is predicted over a normalized time $t \in [0, 1]$, conditioned on $c$.

Training minimizes the discrepancy between the predicted velocity $\hat{v}$ and the ground-truth velocity $u = x_T - x_0$ derived from grasp pairs $(x_0, x_T)$ sampled from FLASH-Drive. The CFM objective, applied at randomly sampled times $t$ and interpolated states $x_t = x_0 + tu$, is given by:

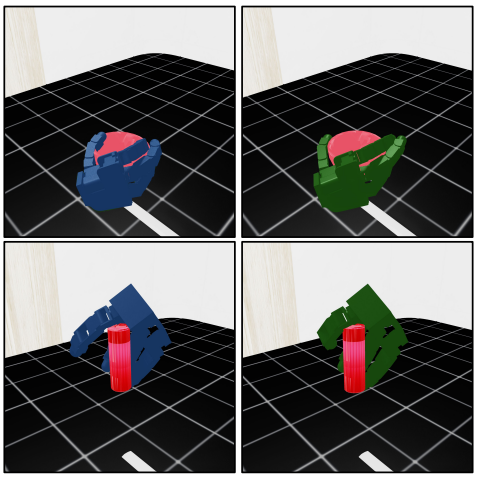


**Figure 2:** Qualitative comparisons on the improvement made on the original grasp quality of the MultiGraspLLM dataset. Our grasps are in green while MultiGraspLLM's are in blue. Our grasps curl around the object and increase the surface in contact with the object.

$$\mathcal{L}_{\text{CFM}}(\theta) = \mathbb{E}_{(x_0, x_T, c)\sim D}\, \mathbb{E}_{t\sim U[0,1]}\, \|f_\theta(x_t, t, c) - (x_T - x_0)\|_2^2, \quad (1)$$

where $f_\theta$ is the neural network parameterized by $\theta$, and $D$ is the FLASH-Drive dataset. To further enforce physical realism during generation, this objective is augmented with a penetration penalty.

$\mathcal{L}_{\text{CFM}}$ constitutes the entire loss for our model. This keeps our objective simple, while the nature of our data generation process allows our model to implicitly optimize for physical plausibility. In this way, we avoid the complexity of weighting several loss terms in order to make trade-offs between physical plausibility and semantic alignment with text conditioning.

The velocity prediction $\hat{v}(x_t, t, c)$ is conditioned on a rich set of inputs represented by $c$. Semantic intent is provided by the natural language prompt, tokenized via the unmodified tokenizer of a pre-trained Qwen-2.5 LLM to leverage its learned representations. Object shape information enters through per-point geometric features $f_{O,geom}$ extracted from the object point cloud $P_O \in \mathbb{R}^{N\times 3}$ by a PointNet++ encoder [23]. To enhance generalization to unseen objects, these geometric features are augmented with semantic context; we extract dense visual features using DINOv2 [24] from renderings of the textured object mesh, project them onto $P_O$, concatenate them with $f_{O,geom}$, and process them through a transformer decoder attending to learned queries. This yields a compact sequence of combined semantic and geometric object features $f_{O,sem} \in \mathbb{R}^{L\times D_{sem}}$. Importantly, the model also receives information about the hand's current geometric state during the flow trajectory. This is achieved by generating the hand mesh corresponding to $x_t$ on-the-fly using an efficient batch meshing pipeline, sampling a hand point cloud $P_H(x_t) \in \mathbb{R}^{M\times 3}$, and extracting its geometric features $f_H \in \mathbb{R}^{M\times D_{geom}}$ via PointNet++. This live hand geometry input allows the model to reason explicitly about potential collisions and contact points throughout generation, overcoming the narrow information bottleneck associated with relying solely on the parameter vector $x_t$. For ablations on semantic features and dynamic updates to the hand point cloud, please refer to Section 3.4.

These conditioning inputs (text embeddings, object features $f_{O,sem}$, live hand features $f_H$, and time $t$) are processed by our network, which employs the Qwen-2.5 LLM architecture as its backbone, ultimately predicting the 25-dimensional velocity vector $\hat{v}$ (see Figure 1b)

### 2.3 Inference

To synthesize a grasp at inference time, given a text prompt and an object point cloud (providing conditioning $c$), FLASH starts from a predefined initial hand state $x_0$ [Placeholder: Specify $x_0$, e.g., canonical pose, random noise]. It then simulates the learned dynamics by numerically integrating the predicted velocity field $\hat{v}(x_t, t, c)$ from $t = 0$ to $t = 1$, using an Dopri9 ODE solver. At each integration step, the model's prediction $\hat{v}$ is conditioned not only on the static inputs but also on the live hand geometry $P_H(x_t)$ derived from the hand state $x_t$ estimated for that step. This continuous feedback loop ensures the generated trajectory remains geometrically aware. The resulting state at $t = 1$, $x_T$, is the final predicted grasp configuration $x^*$.

## 3 Experiments

### 3.1 Experimental Setup

**Dataset and Refinement** – All models are trained and evaluated on FLASH-Drive, our enhanced version of the MultiGraspLLM dataset [4]. As detailed in Section 2.1, FLASH-Drive addresses limitations in original mesh quality by providing high-fidelity watertight object meshes. Using these meshes and our efficient vectorized mesh processing pipeline, we refined the original grasp poses via SDF-based optimization to significantly reduce penetration and improve physical plausibility while preserving functional intent. Furthermore, we augmented the dataset by generating over 200 000 additional structured text annotations using OpenAI's o4-mini model, describing grasps at low, mid, and high levels of abstraction to improve semantic understanding and generalization. This resulted in FLASH-Drive, a large-scale dataset featuring refined grasps across multiple hand embodiments, paired with rich textual descriptions. A comparison with other public datasets is provided contextually in Table 3. Our refinement process demonstrably reduced grasp penetration and improved simulated success rates compared to the original dataset grasps, establishing a higher quality foundation for training.

**Table 1:** Simulation results on **seen** objects (metrics aggregated over test set). Lower is better for CD and Max Pen Dist.; higher is better for Succ. Rate and GPT Score.

| Method | Chamfer Dist. ↓ | Max Pen Dist. ↓ | Succ. Rate ↑ (%) | GPT Score (Align/Feas.) ↑ |
|---|---|---|---|---|
| MultiGraspLLM [4] | **0.37** | 1.04 | **31.98** | – / – |
| DexGraspNet [2] | 0.62 | 1.27 | – | – / – |
| **FLASH** (ours) | 0.43 | **0.36*** | 31.34 | **55.2 / 79.0**** |

*Max Pen Dist. for FLASH measured on generated grasps, may differ slightly from dataset refinement target.
**GPT Scores are dataset averages from FLASH-Drive annotations (Sec. 3.3), indicative of model target.

**Baselines** – We compare FLASH against key prior works relevant to language-conditioned dexterous grasping. MultiGraspLLM [4], the source of our initial dataset, serves as a primary baseline representing recent LLM-based approaches. Additionally, we compare against DexGraspNet [2], a large-scale generative model, representing geometry-centric methods; comparisons are based on geometric metrics evaluated on its generated grasps for comparable objects.

**Metrics** – We evaluate performance using several metrics assessing both physical validity and semantic correctness. Physical plausibility is measured primarily by the Maximum Penetration Distance (Max Pen Dist., ↓) derived from SDFs (Table 1) and an aggregate SDF Loss (↓) used in ablations (Table 4), where lower values indicate less interpenetration. Geometric similarity to ground truth is assessed using Chamfer Distance (CD, ↓) (Table 1, Table 4). Functional success is quantified by the Simulation Success Rate (Succ. Rate, ↑), the percentage of grasps successfully lifting and holding objects in simulation via a standardized heuristic (Section 3.1). Semantic alignment with the text prompt and overall visual physical feasibility are assessed using GPT Scores (↑) (Table 1, Table 2), obtained by prompting OpenAI's o4-mini model with grasp renderings and the corresponding text prompt.

**Simulation Evaluation Heuristic** – Our grasp evaluation approach leverages the IsaacLab simulator, employing a heuristic similar to that of DexGraspNet [2]. This procedure begins with the hand in a pre-grasp pose (flat hand), then moves the hand towards the object based on predicted wrist pose $(T, R)$. Subsequently, a motion plan brings the hand joints towards the target configuration $\theta$. Finally, the hand attempts to lift the object vertically. Grasp success is determined by checking if the object is lifted above a threshold height and remains stable (minimal velocity) after a short duration. This evaluation heuristic allows for scalable testing of numerous grasp candidates efficiently. An example simulation setup is depicted in Figure 4.

**Implementation Details** – We primarily use the Allegro Hand within the IsaacLab simulator for simulation results presented in the main paper, and the LEAP hand for real-world demonstrations. Our FLASH architecture utilizes the Qwen-2.5 (0.5B) LLM backbone and a PointNet++ [23] encoder pre-trained for part segmentation on ShapeNet [25]. Semantic features are derived from DINOv2 [24]. Models were trained on a single NVIDIA H200 GPU for approximately 6 hours.

### 3.2 Zero-Shot Generalization to Unseen Prompts

A crucial capability for language-conditioned models is generalizing to novel instructions. We qualitatively tested FLASH's zero-shot performance by providing novel, complex language instructions for familiar objects seen during training. Examples include prompts specifying unusual contact points or detailed configurations (e.g., "pinch the crown of the hat", "thumb holds the frame of the pistol with the rest on the trigger"). As illustrated in Figure 6, FLASH demonstrates a strong qualitative ability to interpret these unseen instructions and generate geometrically plausible grasps that reflect the specified semantic intent. For instance, it correctly positions fingers according to detailed descriptions, showcasing the benefit of leveraging the LLM's understanding and the augmented vocabulary in FLASH-Drive. While quantitative evaluation across diverse unseen prompts remains future work, these qualitative results suggest promising generalization capabilities stemming from the model's architecture and training data. We did not perform quantitative analysis or comparisons on unseen objects for this work. We also explore FLASH's behavior when fed object point clouds from unseen meshes.

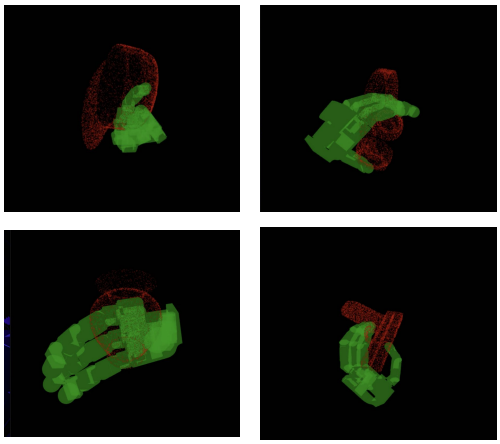


**Figure 3:** Qualitative examples of FLASH generating grasps for **unseen prompts** on seen objects. **Top-left**: "pinch the crown of the hat with all fingers." **Top-right**: "Touch the right ear-cap with thumb and grasp the left ear-cap with the rest." **Bottom-left**: "grasp the handle". **Bottom-right**: "thumb holds the frame of the pistol with the rest on the trigger". Grasps visually align with the novel instructions.

**Figure 4:** IsaacLab simulation evaluation setup with Allegro hands performing diverse grasp attempts on two object types: a cup (left) and a bowl (right), illustrating the environment for heuristic evaluation.

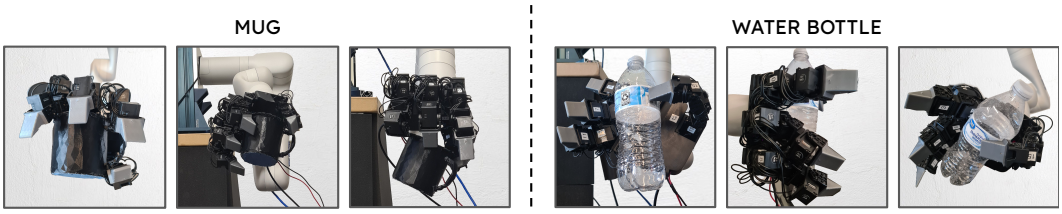



**Figure 5:** Real-world demonstration setup using a LEAP hand mounted on a Kinova Gen 3 arm in a laboratory environment, used for qualitative validation.

### 3.3 Dataset Quality and Annotation Evaluation

The quality of the training data is paramount. As described in Section 2.1, FLASH-Drive was created by refining grasps from MultiGraspLLM and significantly augmenting text annotations using OpenAI's o4-mini model. We prompted the LLM with grasp images and original contact information to generate structured annotations at low, mid, and high levels of detail, captured via function calling with a JSON schema. We evaluated these generated annotations for quality and feasibility using the same LLM (o4-mini). The overall statistics across FLASH-Drive show an average text alignment score of 55.18 and an average physical feasibility score of 79.03 (both out of 100), suggesting the annotations generally capture reasonable intents and correspond to plausible grasps. Per-object statistics, exemplified in Table 2, show variability based on object complexity; for instance, thin objects like knives pose challenges reflected in lower scores. This detailed, multi-level annotation process provides richer supervisory signals for training language-conditioned models like FLASH.

### 3.4 Ablation Studies

To validate our key design choices, we performed ablation studies evaluating variants of FLASH on the seen object set, focusing on geometric quality metrics summarized in Table 4. Removing the continuously updated hand point cloud feedback ('w/o Live Hand PC') significantly degraded

**Table 2:** Example per-object GPT score statistics from FLASH-Drive annotation evaluation (Scores 0–100). High variability exists, e.g., knife scores are lower potentially due to thin geometry challenges.

| | GPT Text Alignment Score (↑) | | | GPT Physical Feasibility Score (↑) | | |
|---|---|---|---|---|---|---|
| **Object Key** | **Count** | **Mean** | **Std.** | **Count** | **Mean** | **Std.** |
| can_s105 | 128 | 64.96 | 12.99 | 128 | 86.41 | 4.52 |
| knife_s225 | 67 | 33.66 | 13.50 | 67 | 56.19 | 21.43 |
| cylinder_bottle_s316 | 127 | 58.03 | 15.09 | 127 | 84.31 | 5.20 |

**Table 3:** Comparison of selected publicly available language-annotated dexterous grasp datasets. FLASH-Drive builds upon MultiGraspLLM with refined geometry and significantly expanded language annotations.

| **Dataset** | **# Grasps (approx)** | **# Objects** | **Text Data Scale** | **Hands** | **Language Source** |
|---|---|---|---|---|---|
| DexGraspNet [2] | 1.3M | 5k+ | Affordance Labels | Shadow, Allegro | Affordance-based |
| MultiGraspLLM [4] | 270k | 2090 | ∼270k | Shadow, Allegro, etc. | GPT-4V (initial) |
| **FLASH-Drive (Ours)** | **270k*** | **2090** | **∼1M**** | **Shadow, Allegro, etc.** | **o4-mini (refined+augmented)** |

*Grasp poses based on MultiGraspLLM, refined for physical plausibility.
**Total text data scale approx. 4x MultiGraspLLM, generated via o4-mini across multiple description levels per grasp.

**Table 4:** Ablation study results on seen objects. Lower SDF Loss and CD indicate better geometric quality.

| **Variant** | **SDF Loss (↓)** | **Chamfer Dist. (CD) (↓)** |
|---|---|---|
| Full FLASH | 0.43 | 0.36 |
| w/o Live Hand PC | 0.66 | 0.44 |
| w/o LLM (transformer from scratch) | 0.37 | 0.35 |
| w/o Refinement (Train on Orig. Data) | 0.64 | 0.35 |
| w/o Semantics (DINOv2 features) | 0.57 | 0.31 |
| w/o Lang (No text conditioning) | 0.49 | 0.35 |

geometric quality (higher SDF Loss and CD), confirming the importance of live geometric reasoning during generation. Replacing the pre-trained Qwen-2.5 backbone with a transformer trained from scratch (‘w/o LLM‘) resulted in slightly different geometric metrics but primarily impacts semantic understanding (discussed qualitatively in Sec. 3.2), which is the LLM’s main role. Training on the original, unrefined dataset grasps (‘w/o Refinement‘) led to significantly worse geometric outcomes (higher SDF Loss), highlighting the benefit of the FLASH-Drive refinement process. Removing the DINOv2 semantic features (‘w/o Semantics‘) also resulted in poorer geometric performance, suggesting these features provide useful context. Finally, omitting language conditioning altogether (‘w/o Lang‘) yielded worse geometric performance compared to the full model, indicating that language conditioning synergistically helps constrain the generation towards more valid configurations present in the language-aligned data. These ablations collectively underscore the benefits of integrating live geometric awareness, leveraging semantic features, utilizing a refined dataset, and incorporating language conditioning via pre-trained LLMs.

### 3.5 Real-World Robotic Demonstration

Finally, to assess FLASH’s applicability in the real world, we conducted qualitative experiments using a Kinova Gen3 robot arm equipped with a LEAP hand [26], shown in Figure 5. We selected several common household objects (e.g., mug, water bottle, drill) and provided text prompts representing typical functional intents (e.g., "grasp the handle to lift", "pick up the bottle by the body"). Grasps were generated offline by FLASH, assuming access to the object mesh, and then executed open-loop on the robot system by moving to the predicted pose and closing fingers to the predicted joint angles. Qualitatively, FLASH successfully generated functionally appropriate and physically stable grasps across the tested objects and prompts. The robot was observed to securely grasp items according to the instructions, for example, correctly using a power grasp for the drill handle when prompted versus attempting a different grasp if only asked to ’pick up’. While these demonstrations are qualitative and do not involve closed-loop control, they strongly suggest that grasps generated by FLASH can transfer effectively to physical hardware and follow nuanced language instructions in practice. We also observed qualitatively that language conditioning significantly influenced the grasp strategy

**Figure 6:** A sample flow trajectory from FLASH when prompted with "grasp the knife with all fingers"

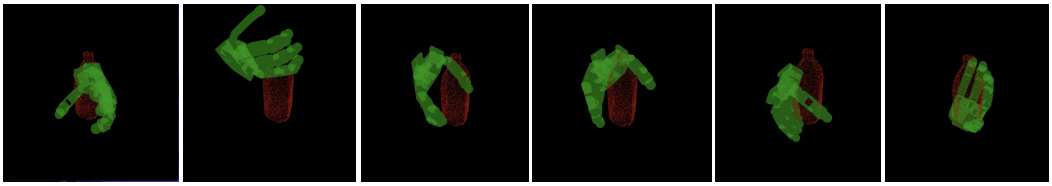


**Figure 7:** A sample trajectory from a diffusion model (used for comparison/ablation visualization), when prompted to "grab the body of the cylindrical bottle". Diffusion processes often explore more diverse states during generation compared to flow matching.

towards the intended function compared to a non-conditioned variant which often defaulted to more generic grasps.

## 4 Conclusion

In this work we present FLASH, a text-conditioned, geometry aware, robot grasping model. Alongside it, we also release FLASH-Drive, a multi-embodiment robot grasping dataset with contact annotations of varying levels of details and improved contact quality. By feeding back a reconstructed hand point cloud back to FLASH as its flowing the hand parameters, we are able to generate higher quality grasps and generalize to new object geometries while not preserving quick inference speed.

## 5 Limitations

Despite promising results, FLASH has limitations reflecting the challenges in conditional grasp synthesis. First, geometric accuracy and input requirements pose difficulties. Performance degrades on thin structures due to Signed Distance Function (SDF) limitations, and the current reliance on complete object point clouds hinders application with partial sensor data or vision-only inputs.

Second, inference speed is a constraint. Generating grasps via iterative ODE solving is computationally intensive, currently limiting real-time use and involving a trade-off between generation speed and final grasp quality. Third, generalization capabilities are bounded by the training data. While FLASH-Drive is large, its object diversity primarily covers a limited set of roughly 40 semantic categories, potentially restricting generalization to truly novel object types. Similarly, robustness to natural language commands significantly diverging from the structured prompts seen during training requires further investigation.

Finally, the focus on static grasp generation means limited consideration of the broader task context. Robust sim-to-real transfer needs further development beyond current qualitative demonstrations, and factors like post-grasp stability under load or suitability for subsequent manipulation steps are not explicitly modeled. Addressing these challenges—improving geometric handling, accelerating inference, broadening generalization, bridging the sim-to-real gap, and incorporating task context—are key directions for future work.

## References


[1] T. Liu, Z. Liu, Z. Jiao, Y. Zhu, and S.-C. Zhu. Synthesizing diverse and physically stable grasps with arbitrary hand structures using differentiable force closure estimator. *IEEE Robotics and Automation Letters*, 7(1):470–477, 2021.

[2] R. Wang, J. Zhang, J. Chen, Y. Xu, P. Li, T. Liu, and H. Wang. Dexgraspnet: A large-scale robotic dexterous grasp dataset for general objects based on simulation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11359–11366. IEEE, 2023.

[3] D. Turpin, T. Zhong, S. Zhang, G. Zhu, E. Heiden, M. Macklin, S. Tsogkas, S. Dickinson, and A. Garg. Fast-grasp'd: Dexterous multi-finger grasp generation through differentiable simulation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8082–8089. IEEE, 2023.

[4] H. Li, W. Mao, W. Deng, C. Meng, H. Fan, T. Wang, P. Tan, H. Wang, and X. Deng. Multi-graspllm: A multimodal llm for multi-hand semantic guided grasp generation, 2025. URL https://arxiv.org/abs/2412.08468.

[5] S. Liu, Y. Zhou, J. Yang, S. Gupta, and S. Wang. Contactgen: Generative contact modeling for grasp generation, 2023. URL https://arxiv.org/abs/2310.03740.

[6] Y. Ye, A. Gupta, K. Kitani, and S. Tulsiani. G-hop: Generative hand-object prior for interaction reconstruction and grasp synthesis, 2024. URL https://arxiv.org/abs/2404.12383.

[7] X. Chang and Y. Sun. Text2grasp: Grasp synthesis by text prompts of object grasping parts, 2024. URL https://arxiv.org/abs/2404.15189.

[8] Z. Zhang, H. Wang, Z. Yu, Y. Cheng, A. Yao, and H. J. Chang. Nl2contact: Natural language guided 3d hand-object contact modeling with diffusion model, 2024. URL https://arxiv.org/abs/2407.12727.

[9] D. Turpin, L. Wang, E. Heiden, Y.-C. Chen, M. Macklin, S. Tsogkas, S. Dickinson, and A. Garg. Grasp'd: Differentiable contact-rich grasp synthesis for multi-fingered hands. In *European Conference on Computer Vision*, pages 201–221, 2022.

[10] P. Li, T. Liu, Y. Li, Y. Geng, Y. Zhu, Y. Yang, and S. Huang. Gendexgrasp: Generalizable dexterous grasping. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8068–8074. IEEE, 2023.

[11] K. Li, J. Wang, L. Yang, C. Lu, and B. Dai. Semgrasp: Semantic grasp generation via language aligned discretization, 2024. URL https://arxiv.org/abs/2404.03590.

[12] OpenAI, J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, R. Avila, I. Babuschkin, S. Balaji, V. Balcom, P. Baltescu, H. Bao, M. Bavarian, J. Belgum, I. Bello, J. Berdine, G. Bernadett-Shapiro, C. Berner, L. Bogdonoff, O. Boiko, M. Boyd, A.-L. Brakman, G. Brockman, T. Brooks, M. Brundage, K. Button, T. Cai, R. Campbell, A. Cann, B. Carey, C. Carlson, R. Carmichael, B. Chan, C. Chang, F. Chantzis, D. Chen, S. Chen, R. Chen, J. Chen, M. Chen, B. Chess, C. Cho, C. Chu, H. W. Chung, D. Cummings, J. Currier, Y. Dai, C. Decareaux, T. Degry, N. Deutsch, D. Deville, A. Dhar, D. Dohan, S. Dowling, S. Dunning, A. Ecoffet, A. Eleti, T. Eloundou, D. Farhi, L. Fedus, N. Felix, S. P. Fishman, J. Forte, I. Fulford, L. Gao, E. Georges, C. Gibson, V. Goel, T. Gogineni, G. Goh, R. Gontijo-Lopes, J. Gordon, M. Grafstein, S. Gray, R. Greene, J. Gross, S. S. Gu, Y. Guo, C. Hallacy, J. Han, J. Harris, Y. He, M. Heaton, J. Heidecke, C. Hesse, A. Hickey, W. Hickey, P. Hoeschele, B. Houghton, K. Hsu, S. Hu, X. Hu, J. Huizinga, S. Jain, S. Jain, J. Jang, A. Jiang, R. Jiang, H. Jin, D. Jin, S. Jomoto, B. Jonn, H. Jun, T. Kaftan, Łukasz

Kaiser, A. Kamali, I. Kanitscheider, N. S. Keskar, T. Khan, L. Kilpatrick, J. W. Kim, C. Kim, Y. Kim, J. H. Kirchner, J. Kiros, M. Knight, D. Kokotajlo, Łukasz Kondraciuk, A. Kondrich, A. Konstantinidis, K. Kosic, G. Krueger, V. Kuo, M. Lampe, I. Lan, T. Lee, J. Leike, J. Leung, D. Levy, C. M. Li, R. Lim, M. Lin, S. Lin, M. Litwin, T. Lopez, R. Lowe, P. Lue, A. Makanju, K. Malfacini, S. Manning, T. Markov, Y. Markovski, B. Martin, K. Mayer, A. Mayne, B. McGrew, S. M. McKinney, C. McLeavey, P. McMillan, J. McNeil, D. Medina, A. Mehta, J. Menick, L. Metz, A. Mishchenko, P. Mishkin, V. Monaco, E. Morikawa, D. Mossing, T. Mu, M. Murati, O. Murk, D. Mély, A. Nair, R. Nakano, R. Nayak, A. Neelakantan, R. Ngo, H. Noh, L. Ouyang, C. O'Keefe, J. Pachocki, A. Paino, J. Palermo, A. Pantuliano, G. Parascandolo, J. Parish, E. Parparita, A. Passos, M. Pavlov, A. Peng, A. Perelman, F. de Avila Belbute Peres, M. Petrov, H. P. de Oliveira Pinto, Michael, Pokorny, M. Pokrass, V. H. Pong, T. Powell, A. Power, B. Power, E. Proehl, R. Puri, A. Radford, J. Rae, A. Ramesh, C. Raymond, F. Real, K. Rimbach, C. Ross, B. Rotsted, H. Roussez, N. Ryder, M. Saltarelli, T. Sanders, S. Santurkar, G. Sastry, H. Schmidt, D. Schnurr, J. Schulman, D. Selsam, K. Sheppard, T. Sherbakov, J. Shieh, S. Shoker, P. Shyam, S. Sidor, E. Sigler, M. Simens, J. Sitkin, K. Slama, I. Sohl, B. Sokolowsky, Y. Song, N. Staudacher, F. P. Such, N. Summers, I. Sutskever, J. Tang, N. Tezak, M. B. Thompson, P. Tillet, A. Tootoonchian, E. Tseng, P. Tuggle, N. Turley, J. Tworek, J. F. C. Uribe, A. Vallone, A. Vijayvergiya, C. Voss, C. Wainwright, J. J. Wang, A. Wang, B. Wang, J. Ward, J. Wei, C. Weinmann, A. Welihinda, P. Welinder, J. Weng, L. Weng, M. Wiethoff, D. Willner, C. Winter, S. Wolrich, H. Wong, L. Workman, S. Wu, J. Wu, M. Wu, K. Xiao, T. Xu, S. Yoo, K. Yu, Q. Yuan, W. Zaremba, R. Zellers, C. Zhang, M. Zhang, S. Zhao, T. Zheng, J. Zhuang, W. Zhuk, and B. Zoph. Gpt-4 technical report, 2024. URL https://arxiv.org/abs/2303.08774.

[13] H. Jiang, S. Liu, J. Wang, and X. Wang. Hand-object contact consistency reasoning for human grasps generation, 2021. URL https://arxiv.org/abs/2104.03304.

[14] Z. Zhao, M. Qi, and H. Ma. Decomposed vector-quantized variational autoencoder for human grasp generation, 2024. URL https://arxiv.org/abs/2407.14062.

[15] S. Huang, Z. Wang, P. Li, B. Jia, T. Liu, Y. Zhu, W. Liang, and S.-C. Zhu. Diffusion-based generation, optimization, and planning in 3d scenes, 2023. URL https://arxiv.org/abs/2301.06015.

[16] Z. Weng, H. Lu, D. Kragic, and J. Lundell. Dexdiffuser: Generating dexterous grasps with diffusion models, 2024. URL https://arxiv.org/abs/2402.02989.

[17] J. Lu, H. Kang, H. Li, B. Liu, Y. Yang, Q. Huang, and G. Hua. Ugg: Unified generative grasping, 2024. URL https://arxiv.org/abs/2311.16917.

[18] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. Lora: Low-rank adaptation of large language models, 2021. URL https://arxiv.org/abs/2106.09685.

[19] F. Zhang and M. Gienger. Affordance-based robot manipulation with flow matching, 09 2024.

[20] A. Tong, N. Malkin, K. Fatras, L. Atanackovic, Y. Zhang, G. Huguet, G. Wolf, and Y. Bengio. Simulation-free schrödinger bridges via score and flow matching, 2024. URL https://arxiv.org/abs/2307.03672.

[21] 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, 2024. URL https://arxiv.org/abs/2302.00482.

[22] B. Hui, J. Yang, Z. Cui, J. Yang, D. Liu, L. Zhang, T. Liu, J. Zhang, B. Yu, K. Dang, et al. Qwen2. 5-coder technical report. *arXiv preprint arXiv:2409.12186*, 2024.

[23] 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*, 30, 2017.

[24] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *Transactions on Machine Learning Research Journal*, pages 1–31, 2024.

[25] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu. Shapenet: An information-rich 3d model repository. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[26] K. Shaw, A. Agarwal, and D. Pathak. Leap hand: Low-cost, efficient, and anthropomorphic hand for robot learning. *Robotics: Science and Systems (RSS)*, 2023.