# Appendix

## A. Pseudo Code of EvolvingGrasp

Pseudo Code of EvolvingGrasp is shown in Algorithm 1 and 2.

**Algorithm 1** Physics-Aware Sampling and Handpose-Wise Preference Optimization

```
Require: Number of inference timesteps $T$, number of finetuning epochs $E_{ft}$, number of objects $K$, physical-aware consistency model $\epsilon_\theta$, test batchsize $B$, test time sequences $S \in \{\tau_i \mid \tau_0 = 0, \tau_{N-1} = T, \tau_i < \tau_{i+1}$ for $i = 0, 1, \ldots, N-1\}$, differentiable functions $c_{skip}$ and $c_{out}$, gradient guidance weight $\{\gamma_i\}_{i=1}^m$.
Copy the parameters of consistency model $\epsilon_{\text{ref}} = \epsilon_\theta$ and set $\epsilon_{\text{ref}}$ to have requires_grad = False.
for $e = 1 : E_{ft}$ do
    # Sample grasping poses
    for $k = 1 : K$ do
        Choose an object $O_k$ and sample $x_T \sim \mathcal{N}(0, \mathbf{I})$
        for $i = 1 : B$ do
            for $n = N - 1 : 0$ do
                $F_\theta(x^i_{k,\tau_n}, \tau_n, O_k) = \frac{1}{\sqrt{\bar{\alpha}_{\tau_n}}} \left( x^i_{k,\tau_n} - \sqrt{1 - \bar{\alpha}_{\tau_n}} \epsilon_\theta \left( x^i_{k,\tau_n,}, \tau_n, O_k \right) \right)$
                $f_\theta(x^i_{k,\tau_n}, \tau_n, O_k) = c_{\text{skip}}(\tau_n) x^i_{k,\tau_n} + c_{\text{out}}(\tau_n) F_\theta(x^i_{k,\tau_n}, \tau_n, O_k)$
                # Sampling with Gradient Guidance:
                $\hat{\mu}_\theta(x^i_{k,\tau_n}, \tau_n, O_k) = \sqrt{\bar{\alpha}_{\tau_{n-1}}} f_\theta(x^i_{k,\tau_n}, \tau_n, O_k) + \sum_{i=1}^m \gamma_i \nabla_{x_{\tau_n}} L_{PA_i} \left( F_\theta(x^i_{k,\tau_n}, \tau_n), \epsilon_\theta \right)$
                $\sigma_{\tau_n} = \sqrt{1 - \bar{\alpha}_{\tau_{n-1}}}$
                $x^i_{k,\tau_{n-1}} = \hat{\mu}_\theta(x^i_{k,\tau_n}, \tau_n, O_k) + \sigma_{\tau_n} \epsilon, \quad \epsilon \sim \mathcal{N}(0, \mathbf{I})$
            end for
        end for
        # Select the Preferred Grasp Poses
        for $i = 0 : B$ do
            if $x^i_0$ grasp object $O_k$ matches human preference then
                $h_i = 1$
            else
                $h_i = -1$
            end if
        end for
        # Efficiently Feedback-driven Finetuning
        for $n = N - 1 : 0$ do
            # Utilizing Fewer Timesteps for Preference Alignment.
            for $i = 1 : B$ do
                with grad:
                $\mu_\theta(x^i_{k,\tau_n}, \tau_n, O_k) = \sqrt{\bar{\alpha}_{\tau_{n-1}}} f_\theta(x^i_{k,\tau_n}, \tau_n, O_k)$, $\mu_{\text{ref}}(x^i_{k,\tau_n}, \tau_n, O_k) = \sqrt{\bar{\alpha}_{\tau_{n-1}}} f_{\text{ref}}(x^i_{k,\tau_n}, \tau_n, O_k)$
                $\pi_\theta \left( x^i_{k,\tau_{n-1}} \mid x^i_{k,\tau_n}, O_k \right) = \frac{1}{\sqrt{2\pi}\sigma_{\tau_n}} \exp(-\frac{(x^i_{k,\tau_n} - \mu_\theta(x^i_{k,\tau_n}, \tau_n, O_k))^2}{2\sigma^2_{\tau_n}})$
                $\pi_{\text{ref}} \left( x^i_{k,\tau_{n-1}} \mid x^i_{k,\tau_n}, O_k \right) = \frac{1}{\sqrt{2\pi}\sigma_{\tau_n}} \exp(-\frac{(x^i_{k,\tau_n} - \mu_{\text{ref}}(x^i_{k,\tau_n}, \tau_n, O_k))^2}{2\sigma^2_{\tau_n}})$
            end for
            Update $\theta$ using gradient descent with LoRA:

                $\nabla_\theta \log \sigma(\sum_{i=1}^{B} h_i \beta \log \frac{\pi_\theta(x^i_{k,\tau_{n-1}} | x^i_{k,\tau_n}, \tau_n, O_k)}{\pi_{\text{ref}}(x_{k,\tau_{n-1}} | x^i_{k,\tau_n}, \tau_n, O_k)})$

        end for
    end for
end for
```

**Algorithm 2** Physical-Aware Distillation

```
Require: Training dataset $D_t$, number of training epochs $E_t$, learning rate $\eta$, pre-trained diffusion model $\epsilon_\theta$, number of timesteps $T_{dm}$, distance metric $d(\cdot,\cdot)$, EMA rate $\mu$, noise schedule $\{\alpha_t\}_{t=1}^{T_{dm}}$, physics-aware constraints weights $\{\alpha_i\}_{i=1}^{m}$.
Copy the parameters of the pre-trained diffusion model as the target network $\epsilon_{\theta'} = \epsilon_\theta$
for $e = 1 : E$ do
    for $k = 1 : K$ do
        Choose an object $O_k$ and sample $x_0 \sim D_t$, $n \sim \mathcal{U}[1, N]$
        Sample $x_{\tau_n} \sim \mathcal{N}(\sqrt{\bar{\alpha}_{\tau_n}} x_0, (1 - \bar{\alpha}_{\tau_n})\mathbf{I})$
        $F_\theta(x_{\tau_n}, \tau_n, O_k) = \frac{1}{\sqrt{\bar{\alpha}_{\tau_n}}} \left(x_{\tau_n} - \sqrt{1 - \bar{\alpha}_{\tau_n}} \epsilon_\theta\left(x_{\tau_n}, \tau_n, O_k\right)\right)$
        $\hat{x}^*_{\tau_{n-1}} = \sqrt{\bar{\alpha}_{\tau_{n-1}}} F_\theta(x_{\tau_n}, \tau_n, O_k) + \sqrt{1 - \bar{\alpha}_{\tau_{n-1}}} \epsilon, \quad \epsilon \sim \mathcal{N}(0, \mathbf{I})$
        $\mathcal{L}_{PAD} = \mathbb{E}\left[d\left(f_\theta\left(x_{\tau_n}, \tau_n\right), f_{\theta'}\left(\hat{x}^*_{\tau_{n-1}}, \tau_{n-1}\right)\right)\right] + \sum_{i=1}^{m} \alpha_i L_{PA_i}\left(F_\theta(x_{\tau_n}, \tau_n), \epsilon_\theta\right)$
        $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}_{PAD}$
        $\theta' \leftarrow \text{stopgrad}\left(\mu\theta' + (1 - \mu)\theta\right)$
    end for
end for
```

## B. Proof

Defined on the new path, the proof of the upper bound is as follows:

$$
\begin{aligned}
L_{\mathrm{BT}}(\theta) &= -\mathbb{E}_{x_0^{w,l} \sim \mathcal{D}} \log \sigma \left( \underset{x_{1:T}^{w,l} \sim \pi_\theta(x_{1:T}^{w,l} | x_0^{w,l})}{\beta \mathbb{E}} \left[ \log \frac{\pi_\theta(x_{0:T}^w)}{\pi_{\mathrm{ref}}(x_{0:T}^w)} - \log \frac{\pi_\theta(x_{0:T}^l)}{\pi_{\mathrm{ref}}(x_{0:T}^l)} \right] \right) \\
&= -\mathbb{E}_{x_0^{w,l} \sim \mathcal{D}} \log \sigma \left( \underset{x_{1:T}^{w,l} \sim \pi_\theta(x_{1:T}^{w,l} | x_0^{w,l})}{\beta \mathbb{E}} \left[ \sum_{n=1}^{N} \log \frac{\pi_\theta\left(x_{\tau_{n-1}}^w \mid x_{\tau_n}^w\right)}{\pi_{\mathrm{ref}}\left(x_{\tau_{n-1}}^w \mid x_{\tau_n}^w\right)} - \sum_{n=1}^{N} \log \frac{\pi_\theta\left(x_{\tau_{n-1}}^l \mid x_{\tau_n}^l\right)}{\pi_{\mathrm{ref}}\left(x_{\tau_{n-1}}^l \mid x_{\tau_n}^l\right)} \right] \right) \\
&= -\mathbb{E}_{x_0^{w,l} \sim \mathcal{D}} \log \sigma \left( \underset{x_{1:T}^{w,l} \sim \pi_\theta(x_{1:T}^{w,l} | x_0^{w,l})}{\beta \mathbb{E}} N \mathbb{E}_n \left[ \log \frac{\pi_\theta\left(x_{\tau_{n-1}}^w \mid x_{\tau_n}^w\right)}{\pi_{\mathrm{ref}}\left(x_{\tau_{n-1}}^w \mid x_{\tau_n}^w\right)} - \log \frac{\pi_\theta\left(x_{\tau_{n-1}}^l \mid x_{\tau_n}^l\right)}{\pi_{\mathrm{ref}}\left(x_{\tau_{n-1}}^l \mid x_{\tau_n}^l\right)} \right] \right) \\
&= -\mathbb{E}_{x_0^{w,l} \sim \mathcal{D}} \log \sigma \left( \underset{n, x_{\tau_n}^{w,l} \sim \pi_\theta\left(x_{\tau_n}^{w,l} | x_0^{w,l}\right)}{N \beta \mathbb{E}} \left[ \log \frac{\pi_\theta\left(x_{\tau_{n-1}}^w \mid x_{\tau_n}^w\right)}{\pi_{\mathrm{ref}}\left(x_{\tau_{n-1}}^w \mid x_{\tau_n}^w\right)} - \log \frac{\pi_\theta\left(x_{\tau_{n-1}}^l \mid x_{\tau_n}^l\right)}{\pi_{\mathrm{ref}}\left(x_{\tau_{n-1}}^l \mid x_{\tau_n}^l\right)} \right] \right) \\
&\leq - \underset{\substack{x_0^{w,l} \sim \mathcal{D}, t \sim \mathcal{U}(0,T), \\ n, x_{\tau_n}^w \sim \pi_\theta\left(x_{\tau_n}^w | x_0^w\right)}}{\mathbb{E}} \log \sigma \left( \beta \log \frac{\pi_\theta\left(x_{\tau_{n-1}}^w \mid x_{\tau_n}^w\right)}{\pi_{\mathrm{ref}}\left(x_{\tau_{n-1}}^w \mid x_{\tau_n}^w\right)} - \beta \log \frac{\pi_\theta\left(x_{\tau_{n-1}}^l \mid x_{\tau_n}^l\right)}{\pi_{\mathrm{ref}}\left(x_{\tau_{n-1}}^l \mid x_{\tau_n}^l\right)} \right)
\end{aligned}
\tag{18}
$$

where the last inequality is based on Jensen's inequality and $-\log \sigma(\cdot)$ is a strict convex function. Therefore, we use the new objective 18 to optimize the whole model with LoRA [11].

Table 3. Cross-dataset evaluation results. The highest performances are highlighted in **bold**, while the second-highest performances are indicated with underline.

| **Testing Dataset** | DexGraspNet | | | MultiDex | | | RealDex | | | DexGRAB | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Training Dataset** | **Suc.6** ↑ | **Suc.1** ↑ | **Pen.** ↓ | **Suc.6** ↑ | **Suc.1** ↑ | **Pen.** ↓ | **Suc.6** ↑ | **Suc.1** ↑ | **Pen.** ↓ | **Suc.6** ↑ | **Suc.1** ↑ | **Pen.** ↓ |
| DexGraspNet | 65.2 | 92.7 | 17.2 | 73.4 | 97.1 | **9.7** | **54.1** | **90.1** | **19.4** | 58.1 | 94.3 | 20.6 |
| MultiDex | **67.6** | **94.0** | **19.5** | **76.8** | 98.4 | 13.0 | 51.9 | 88.6 | **19.4** | **65.6** | **96.8** | 19.5 |
| RealDex | 52.2 | 81.1 | 20.7 | 51.5 | 88.1 | 14.0 | 50.6 | 82.5 | 20.3 | 46.0 | 80.0 | **18.3** |
| DexGRAB | 64.9 | 92.6 | 17.1 | 75.3 | 99.3 | 9.9 | 53.1 | 88.5 | 19.7 | 57.7 | 95.2 | 23.7 |

# C. Details of Experimental Setup

## C.1. Dataset Setups

DexGraspNet is a large-scale dataset for dexterous grasping, comprising 1.32 million grasp samples across 5,355 objects from 133 diverse categories. While its optimization-based generation ensures high quality and diversity, its applicability in real-world scenarios is limited.

In contrast, MultiDex focuses on a smaller set of 58 everyday objects but offers a rich variety of grasping poses for each object. This makes it an ideal dataset for studying the diversity of grasping configurations and developing methods that can generate a wide range of effective grasps for common objects.

Realdex shifts the focus to real-world applications by capturing natural human grasping behaviors. It contains 59,000 samples across 52 objects, making it highly suitable for training robots to learn human-like grasping poses. Although it covers fewer object categories, its real-world grounding allows it to effectively validate the generalization and practicality of dexterous grasping methods in real environments.

DexGRAB, derived from human hand interaction data, provides over 1.64 million grasp samples across 51 distinct objects. It offers rich grasping patterns and natural interaction behaviors, making it a valuable resource for understanding human grasping strategies. Similar to DexGraspNet, DexGRAB's data quality is high after filtering, but its real-world applicability may also face some limitations due to its primarily simulation-based nature.

Together, these datasets offer a range of strengths and limitations, from the large-scale optimization-based approaches of DexGraspNet and DexGRAB to the real-world grounding of Realdex and the diversity-focused MultiDex. Each dataset contributes unique insights and challenges to the field of dexterous grasping research.

## C.2. Technical Details

Our EvolvingGrasp contains distillation and sampling, which are implemented using PyTorch [28] platform in one NVIDIA Tesla A40 GPU. In the distillation process, we train EvolvingGrasp for 1,000 epochs with a batch size of 1,200. During both the distillation and preference finetuning processes, the initial learning rate is set to 0.00001. For the distillation process, we use the pretrained checkpoint of each dataset in Dexgrasp Anything [49] and the learning rate remains unchanged. During inference, the success rate of the generated grasping poses is firstly evaluated. If the success rate improves, the learning rate is adaptively reduced, otherwise, it is increased accordingly. Additionally, the adjustment of the learning rate is constrained within a predefined threshold range to ensure it remains within reasonable bounds. The sampling and preference optimization processes are implemented in test split of each corresponding dataset.

Table 4. Evaluating Cross-Dataset Generalization. Model performance is compared on RealDex, with training on DexGraspNet.

| **Method** | **Suc.6** ↑ | **Suc.1** ↑ | **Pen.** ↓ |
|---|---|---|---|
| SceneDiffuser | 16.1 | 52.1 | 29.2 |
| GraspTTA | 25.5 | 64.8 | 31.6 |
| UGG | 33.6 | 74.5 | 33.0 |
| DexGrasp Any. | 38.4 | 77.5 | **19.2** |
| Ours w/o HPO | 52.6 | 88.8 | 19.5 |
| Ours | **54.1** | **90.1** | 19.4 |

# D. Additional Experiments

## D.1. Performance of Cross Dataset

We conducted cross-validation experiments on four datasets with our method and one dataset with four methods. The results with four datasets are shown in Table 3, which demonstrate that the Physics-Aware Consistency Model trained on the Multidex dataset achieved the best performance when tested on the other datasets. The model trained and tested on the DexGRAB and DexGraspNet datasets showed moderate performance. Since Realdex is a real-world dataset with relatively lower quality, the performance of the model trained and tested on Realdex was relatively worse. The results with four methods are shown in Table 4, which illustrate that our methods can significantly improve the grasping performance on the realdex dataset compared with other methods.

## D.2. More Ablation Studies

Table 5. Ablation study on different hyperparameters (i.e., the regularization weight $\beta$, the number of iterations per finetuning epoch $N_{ft}$). We report the results under 2, 4, and 8 steps during sampling.

| $T$ | $\beta$ | **Suc.6** ↑ | **Suc.1** ↑ | **Pen.** ↓ | $N_{ft}$ | **Suc.6** ↑ | **Suc.1** ↑ | **Pen.** ↓ |
|---|---|---|---|---|---|---|---|---|
| 2 | 0.1 | 65.6 | 97.5 | 15.2 | 1 | 65.9 | 97.2 | 15.3 |
| | 0.5 | 63.4 | 96.8 | 15.2 | 3 | 63.4 | 97.1 | 15.1 |
| | 1.0 | 65.9 | 97.2 | 15.3 | 5 | 66.2 | 96.9 | 15.2 |
| | 2.0 | 65.9 | 97.2 | 15.3 | 10 | 65.3 | 97.5 | 15.3 |
| 4 | 0.1 | 75.9 | 97.1 | 13.1 | 1 | 76.8 | 98.4 | 13.0 |
| | 0.5 | 77.1 | 97.2 | 13.1 | 3 | 75.9 | 97.8 | 13.0 |
| | 1.0 | 76.8 | 98.4 | 13.0 | 5 | 77.5 | 97.5 | 13.2 |
| | 2.0 | 77.2 | 97.2 | 13.2 | 10 | 75.9 | 97.5 | 13.1 |
| 8 | 0.1 | 79.4 | 97.8 | 12.2 | 1 | 80.3 | 98.7 | 12.3 |
| | 0.5 | 76.8 | 97.8 | 12.1 | 3 | 76.5 | 98.4 | 12.2 |
| | 1.0 | 80.3 | 98.7 | 12.3 | 5 | 78.8 | 98.1 | 12.2 |
| | 2.0 | 78.7 | 97.5 | 12.2 | 10 | 80.0 | 98.1 | 12.2 |

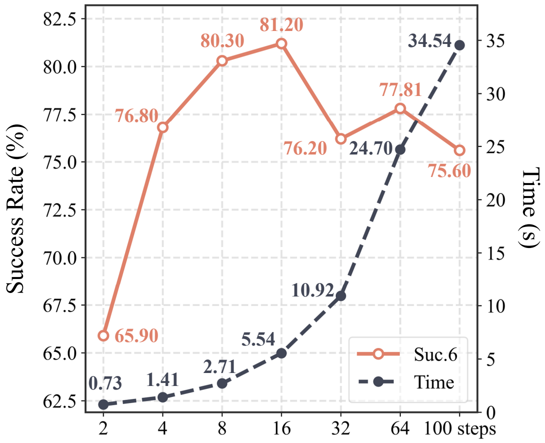

Figure 8. The effect of different sampling steps on grasping performance. The red solid line represents the grasping success rate, while the black dashed line denotes the time consumption.

**Impact of different hyperparameters.** A comprehensive analysis of different hyperparameters (i.e, regularization weight $\beta$, number of finetuning $N_{ft}$ every epoch, number of timesteps $T$) to the performance during preference alignment is reported in Table 5 and Fig. 8. Table 5 demonstrates that when the sampling time steps is relatively small, such as 2 or 4 steps, increasing the number of finetuning iterations $N_{ft}$ and raising the value of the regularization coefficient $\beta$ can enhance the model's performance. Conversely, when the sampling steps is larger, employing fewer $N_{ft}$ and a smaller $\beta$ value helps maintain the model at a high performance level. Fig. 8 shows that as the number of sampling steps increases, the grasping performance first improves and then declines. The highest grasping success rate is achieved when the sampling step is set to 16. The potential reason is that during the multi-step sampling process, each step introduces minor errors in noise handling. These errors may be masked in early steps but accumulate over time, eventually degrading the sample quality.