2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

Adapting for Calibration Disturbances: A Neural Uncalibrated Visual Servoing Policy

Hongxiang Yu[1]†, Anzhe Chen[1]†, Kechun Xu[1], Dashun Guo[1], Yufei Wei[1], Zhongxiang Zhou[1], Xuebo Zhang[2], Yue Wang[1]* and Rong Xiong[1]

***Abstract*— Visual servoing (VS) is a widely used technique in industries where there are hundreds of robots, but it requires accurate camera calibration including camera intrinsic and extrinsic parameters. However, it is labour-intensive to calibrate robots one-by-one in practical use. In this paper, we propose a neural uncalibrated VS policy (NUVS) that can adapt to calibration disturbances with an adaption mechanism and a control-oriented guidance. It bridges the disturbance adaption of classical VS methods and the large convergence of learning-based VS methods. NUVS estimates the calibration embedding from past observations and servos to the desired pose under the supervision of a PBVS that can access the ground truth in simulation. With this adaption mechanism, NUVS outperforms the classical IBUVS algorithm when facing large initial camera pose offsets under the calibration disturbance. Supplementary material in: https://sites.google.com/view/neural-uncalibrated-vs**

I. Introduction

Visual servo (VS) is an essential task for robot control, which guides the robot by comparing the desired image and the current image[1]. As the solution is easy to deploy, and the camera is of low cost, it is popular for object oriented positioning [2], [3].

One of key components of VS accuracy is the calibration, which includes the camera intrinsic parameter, as well as the extrinsic parameter between the camera and the robot tool center point (TCP). Therefore, considering the production scale utilization of VS, the burden of one-by-one calibration becomes labour-intensive. The worse is the calibration may vary by a small external disturbance, or even drift in long term, thus the maintenance of VS based robot is expensive.

Motivated by the challenge of calibration disturbance, efforts have been made in the recent decade. PBVS [4], [5], [6] is globally asymptotically stable [7] and has predicable trajectory in 3D space, but it relies on the calibration-error-sensitive 3D reconstruction [8]. IBVS is less sensitive to disturbance, since its error signal is defined directly in image space. The mainstream methods of IBVS that tackle calibration disturbance are known as Image Based Uncalibrated Visual Servoing (IBUVS) [9], [10], [11], [12]. IBUVS online estimates the image Jacobian matrix for unknown and unstructured environments, adapts to the disturbance of calibration, and modulates the controller for stabilization. However, IBUVS suffers from the drawbacks familiar to IBVS such as Jacobian singularity, unpredictable trajectory in 3D space and feature loss problem [13].

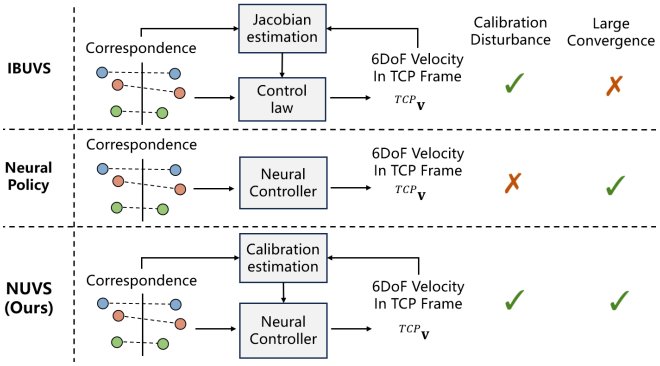

Fig. 1: Classical VS is sensitive to calibration disturbance, while IBUVS can deal with the disturbance but has a limited domain of convergence. Neural policy has large convergence basin, however can not adapt to disturbances. NUVS uses neural networks to estimate the calibration embedding for adaption, and is supervised by PBVS to achieve better servo performance.

With the development of deep learning, recent methods develop image based neural visual servo policy [3], [14]. As PBVS has a more predicable 3D trajectory, it is used to supervise the image based neural controller to achieve larger convergence basin than classical IBVS. But they either require accurately calibrated intrinsic and extrinsic parameters [14] or overfit to a fixed calibration [3]. Therefore, a question is raised: *Can we endow the learning based VS policy with adaption to calibration disturbance?*

As shown in Fig. 1, compared with the classic IBUVS, the key insight is to develop the adaption mechanism to modulate the neural policy, so that the disturbance of calibration can be considered in the loop. Inspired by the recent progress on meta reinforcement learning of drone control [15] and quadruped robot locomotion [16], the key issues are the architecture design and the training strategy.

In this paper, bridging the disturbance adaption of classic VS methods, and the large convergence of learning based VS methods, we propose a neural uncalibrated VS policy (NUVS) with an adaption mechanism and the control oriented guidance. We consider the calibration disturbance both on intrinsic parameters i.e. focal length, principal center, and extrinsic parameters i.e. rotation and translation between the camera and the robot TCP. These calibration parameters are embedded into the latent space, which then modulates the neural VS policy as the network input. When these parameters are unknown in practice, the past observations

This work was supported in part by the National Nature Science Foundation of China (Grant 62173293 and 62373322) and the Innovation and Development Special Fund of the Hangzhou Chengxi Sci-tech Innovation Corridor. Hongxiang Yu, Anzhe Chen, Kechun Xu, Dashun Guo, Yufei Wei, Zhongxiang Zhou, Yue Wang and Rong Xiong are with Zhejiang University, Hangzhou, Zhejiang, China. Xuebo Zhang is with Nankai University, Tianjin, China. Hongxiang Yu, Anzhe Chen contributed equally to this work. Yue Wang is the corresponding author wangyue@iipc.zju.edu.cn

are employed to estimate the current calibration embedding, making the whole policy adaptive. Following the classic adaptive control [17], we do not pursue the accurate prediction of calibration estimation, but the control-oriented accuracy of the whole adaptive policy. We supervise the whole network by the PBVS accessible to the privileged data in the simulation, then regard the calibration embedding as a regularizer. Finally, the NUVS policy demonstrates the superior performance in both simulation and the real world. Overall, the contributions of this paper are three-fold:

- We propose a neural uncalibrated visual servo policy that estimates the calibration embedding, modulates the network, and guides the robot to the desired pose.
- We supervise the neural UVS policy by the PBVS accessible to the ground truth in the simulation, with the latent estimation accuracy being the regularizer.
- We validate the policy in both the simulation and real world experiments. It adapts to calibration disturbance well and demonstrates the superior performance than the comparative methods.

II. Related Works

Classic VS Methods: PBVS [4], [5] relies on the accurate intrinsic and precise 3D model measurement for pose estimation [18]. IBVS [1], [19] designs the controller with features represented on the 2D image plane, which is insensitive to the camera calibration error, but suffers from the feature loss problem due to the complex trajectory in 3D space. [20] tries to solve the feature loss issue by using a Kalman filter to estimate the positions of features when they are lost. However, estimation error will degrade the performance and accumulation of errors may induce unexpected failure. [21], [22] use MPC-based methods to ensure the visibility constraints but suffer heavy computational burden. As for hybrid visual servoing, some methods [23] switch between IBVS and PBVS to prevent the drawback of single controller, which introduce extra difficulties in designing switching conditions manually.

Uncalibrated VS Methods: Qian [24] is the first to achieve general all uncalibrated model-free visual servoing which estimate Jacobian using Kalman-Bucy filter (KBF). Other methods can be categorized into Broyden-Gauss-Newton (BGN) method [25], [26], [27], [28], Broyden recursive least squares (BRLS) method [29] and Broyden population (BP) method [30]. Hao et. al [11] gives a detailed implementation and comparison of these methods and we use the KBF method as IBUVS in our work for comparison. [12] also compares the performance of several methods used for the estimation of an image Jacobian matrix in uncalibrated model-free visual servoing.

Learning Based VS Methods: Recently, there has been a renewed interest in applying neural networks for visual servoing. Pose based methods [31], [2], [32] use neural networks to estimate the relative pose between current pose and desired pose with images. The estimated relative pose is then given to PBVS controller for control. These pose estimators are affected by the variation of camera intrinsic and well calibrated camera extrinsic is also needed to transfer the velocity to TCP frame. Moreover, with a imprecise camera extrinsic, they cannot use the repeat positioning data for self-adaption. Correspondence based methods [33], [34], [35] use neural networks to predict matched 2D visual features or optical flow. The extraction of 2D correspondence won't be affected by camera intrinsic. However, the control command is calculated later through IBVS that has internal deficiency such as small convergence region and local minima [36], [37]. [3], [38], [14] use neural controller to replace traditional VS controller. Supervise by PBVS, they have larger convergence basin. But they either require accurately calibrated intrinsic and extrinsic parameters[14] or overfit to a fixed calibration[3], [38].

III. Method

NUVS is equipped with an adaption mechanism and the control oriented guidance to estimate the calibration embedding, modulate the network, and guide the robot to the desired pose. The whole policy consists of three parts. First, we describe the process of calibration parameters encoding using a variational autoencoder in section III-A. Second, we construct the base policy by combining the frozen encoder from the first part and a learnable neural controller in section III-B. The neural controller is trained with domain randomization in simulation taking current feature correspondence as input and modulating by calibration embedding. Third, we propose an adaptive policy to estimate the calibration embedding of the unknown camera parameters with past observations for modulating the trained neural controller to acquire the robot's velocity.

A. Calibration Embedding Encoder Training

To depict various disturbance of camera calibrations, we sample one million calibration from an uniform distribution given the origin intrinsic K and extrinsic ${}^{TCP}\mathbf{T}_c$, where TCP represents TCP frame. To dilute the training cost of the following base policy and to speed up the training process by the more significant gradient, we project the calibration parameter $\boldsymbol{e} \in \mathbb{R}^7$ into the latent space to get a calibration embedding $\boldsymbol{z}$ that conforms to the normal distribution[39].

Network architecture: In detail, $\boldsymbol{e}$ consists of the rotation axis and angle of extrinsic ${}^{TCP}\mathbf{T}_c$, as well as the focal length and the principal point coordinates of K. In simulation, we simulate intrinsic K with a projection function and simulate the extrinsic ${}^{TCP}\mathbf{T}_c$ with a camera stick on the robot TCP as shown in Fig. 2. This calibration randomization guarantees the generalization of our neural servo policy. Specific sample details are shown in the supplementary material. We pretrain a variational autoencoder (VAE) [39] to encode the $\boldsymbol{e}$ into a latent vector $\boldsymbol{z} \in \mathbb{R}^8$. The encoder δ of VAE is a 3-layer multi-layer perception network (MLP) (32,16,8).

Loss design: As Eq. (1)-(3) shown, the loss function $\mathcal{L}_e$ of VAE consists of two parts: the reconstruction loss $\mathcal{L}_e^{rec}$ and the regular loss $\mathcal{L}_e^{reg}$. The reconstruction loss $\mathcal{L}_e^{rec}$ leads the output of decoder $\hat{\boldsymbol{e}}$ to converge to the input $\boldsymbol{e}$ and the

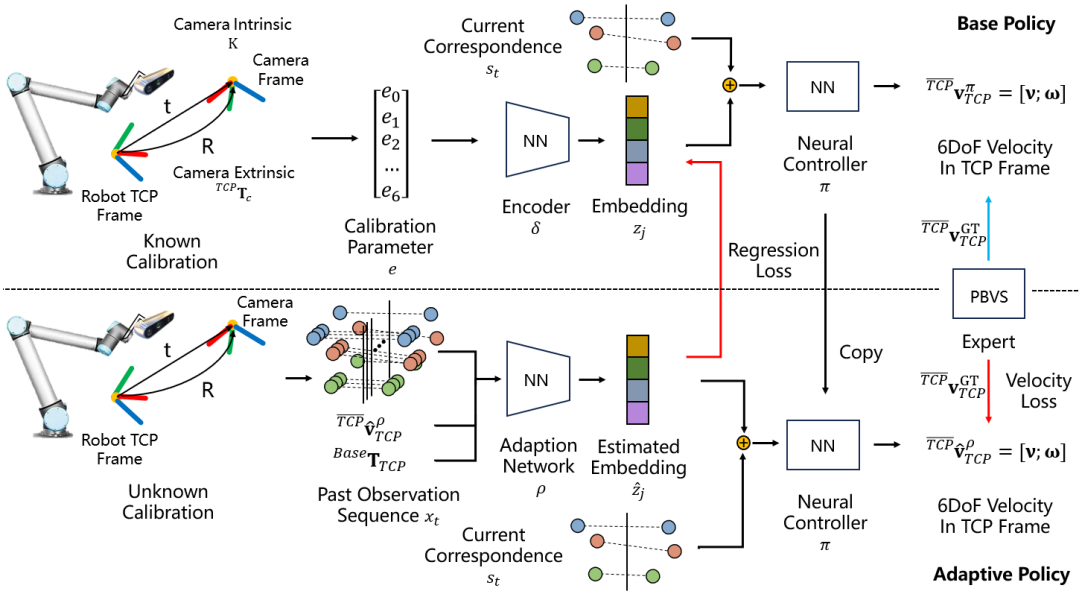

Fig. 2: The method flow for the base policy and adaptive policy. The base policy is modulated by the camera calibration embedding and tries to imitate an expert PBVS controller. The adaptive policy tries to estimate current calibration embedding given the correspondences, controller predictions and TCP's poses in the past 10 steps. The adaptive policy is supervised by the control-oriented loss (velocity loss & regression loss).

regular loss l_e^{reg} empowers the latent vector $\boldsymbol{z}$ conforming to an independent normal distribution.

$$\mathcal{L}_e = \mathcal{L}_e^{rec} + \mathcal{L}_e^{reg} \tag{1}$$

$$\mathcal{L}_e^{rec} = -||\boldsymbol{e} - \hat{\boldsymbol{e}}||_2^2 \tag{2}$$

$$\mathcal{L}_e^{reg} = -\frac{1}{2}\sum_{i=0}^{n}(\mu_i^2 + \sigma_i^2 - log\sigma_i^2 - 1) \tag{3}$$

where the $\boldsymbol{\mu}$, $\boldsymbol{\sigma}$ are the mean and standard deviation of $\boldsymbol{z}$. A reparameterization trick is applied to sample the $\boldsymbol{z}$ as Eq.(5) describes, which makes the sampling process differentiable.

$$\boldsymbol{\mu}, \boldsymbol{\sigma} = \delta(\boldsymbol{e}) \tag{4}$$

$$\boldsymbol{z} = \boldsymbol{\mu} + \boldsymbol{\epsilon} \odot \boldsymbol{\sigma} \tag{5}$$

where $\boldsymbol{\epsilon} \sim \mathcal{N}(0, I_d)$.

B. Base Policy Training

Base policy consists of VAE's encoder δ and a neural controller π to predict robot's velocity. The encoder δ is freezed. When the calibration is disturbed, the controller's output should also be modified. To make π adapt to any calibration disturbance and output the robot's velocity command according to real calibration parameters, we take the encoded intrinsic and extrinsic parameters as part of π's input. In other words, we use the calibration embedding to modulate the neural controller's behaviour, so that it can cope with disturbance and control the robot movement more accurately.

Network architecture: We randomly sample a pair of intrinsic and extrinsic parameters $\boldsymbol{e_j}$ from the calibration distribution to get the embedding with $\boldsymbol{z_j} = \delta(\boldsymbol{e_j})$ and generate corresponding supervision signals based on PBVS. At time t, the input of the neural controller π is the current feature correspondence $\boldsymbol{s_t} \in \mathbb{R}^{2m}$ consisting coordinates of m current and desired feature, in addition with the calibration embedding $\boldsymbol{z_j} \in \mathbb{R}^8$, as shown in the upper part of Fig. 2. π is a 4-layer multi-layer perception network (MLP) (256, 256, 128, 7). It outputs the velocity direction $\mathbf{v}_{\text{dir}}^{\text{pred}} = [\boldsymbol{\nu}_{\text{dir}}^{\text{pred}}; \boldsymbol{\omega}_{\text{dir}}^{\text{pred}}]$ and a scalar l^{pred}:

$$\mathbf{v}_{\text{dir}}^{\text{pred}}, l^{\text{pred}} = \pi(\boldsymbol{s_t}, \boldsymbol{z_j}) \tag{6}$$

Then we can get the actual control:

$$\boldsymbol{\nu}^{\text{pred}} = \frac{\boldsymbol{\nu}_{\text{dir}}^{\text{pred}}}{\|\boldsymbol{\nu}_{\text{dir}}^{\text{pred}}\|_2}\mathcal{T}(l^{\text{pred}}); \quad \boldsymbol{\omega}^{\text{pred}} = \frac{\boldsymbol{\omega}_{\text{dir}}^{\text{pred}}}{\|\boldsymbol{\nu}_{\text{dir}}^{\text{pred}}\|_2}\mathcal{T}(l^{\text{pred}}) \tag{7}$$

where $\mathcal{T}(\cdot) = 1 + \text{ELU}(\cdot)$, and $\mathcal{T}(l^{\text{pred}})$ equals to the velocity norm. $\mathcal{T}$ decays exponentially to zero when input is negative, encouraging the network to predict more accurate velocity when close to the desired pose to avoid damping behavior. Thus we can recover robot TCP velocity ${}^{T\bar{C}P}\mathbf{v}_{TCP}^{\pi} = [\boldsymbol{\nu}^{\text{pred}}; \boldsymbol{\omega}^{\text{pred}}]$, which can be used to control the robot.

Loss design: Since PBVS has a larger convergence basin and more predictable trajectory in Cartesian space, we choose it as the supervision. In order to avoid the feature points moving out of the camera's field of view (FoV), we select the interaction matrix that ensures that the target object stays in the center of the camera's FoV [1] and calculate the velocity ${}^{\bar{c}}\mathbf{v}_c^{\text{GT}} \in \mathbb{R}^6$ in camera frame. As we know the ground truth camera extrinsic in training, we transfer the ${}^{\bar{c}}\mathbf{v}_c^{\text{GT}}$ to ${}^{T\bar{C}P}\mathbf{v}_{TCP}^{\text{GT}}$ with extrinsic as the final supervision. We supervise the direction and the norm of the predicted velocity separately:

$$\begin{aligned}\mathcal{L}_{\text{dir}} &= 1 - \text{cosine\_similarity}({}^{T\bar{C}P}\mathbf{v}_{TCP}^{\pi}, {}^{T\bar{C}P}\mathbf{v}_{TCP}^{\text{GT}}) \\ \mathcal{L}_{\text{norm}} &= \text{MSE}\left(l^{\text{pred}}, \mathcal{T}^{-1}\left(||{}^{T\bar{C}P}\mathbf{v}_{TCP}^{\text{GT}}||_2\right)\right)\end{aligned} \tag{8}$$

The final servo loss as:

$$\mathcal{L}_{\pi} = \mathcal{L}_{\text{dir}} + 0.5\mathcal{L}_{\text{norm}} \tag{9}$$

The base policy is trained by imitation learning with dataset aggregation [40] for faster convergence.

C. Adaptive Policy Training

The base policy acquires ground truth calibration $\boldsymbol{e_j}$ to get $\boldsymbol{z_j}$ as a part of neural controller π's input. However, it's impracticable to obtain the accurate $\boldsymbol{e_j}$ in practical use, a intuitive solution is to propose an adaptive policy that estimates $\hat{\boldsymbol{e}}_{\boldsymbol{j}}$. But our ultimate goal is not to identify system calibration but to obtain a correct action for visual servoing. Inspired by the classic adaptive control[17], we design the adaptive policy to estimate calibration embedding $\hat{\boldsymbol{z}}_{\boldsymbol{j}}$ and minimize the output of $\pi(\boldsymbol{s_t}, \hat{\boldsymbol{z}}_{\boldsymbol{j}})$ with ground truth at the same time. The adaptive policy consists of the adaption network ρ and the controller π with π's parameters freezed.

Network architecture: As the camera calibration process requires dozens of calibration plate poses and TCP poses respect to the robot base frame ${}^{b}\mathbf{T}_{TCP}$, we set the input of ρ to be the past observation sequence $\boldsymbol{x}_t$. $\boldsymbol{x}_t \in \mathbb{R}^{(2m+7+7)\times k}$ consists of past feature correspondence $\boldsymbol{s}_{t-k+1:t} \in \mathbb{R}^{2m\times k}$, past actions $\boldsymbol{a}_{t-k+1:t} \in \mathbb{R}^{7\times k}$ and the past k TCP poses ${}^{b}\mathbf{T}_{TCPt-k+1:t}$. To prevent ρ from overfitting on the absolute TCP poses, the sequence of TCP poses are regularized by dividing the first pose ${}^{b}\mathbf{T}_{TCPt-k+1}$ to get the relative movement. ${}^{b}\mathbf{T}_{TCP}$ are convert to the vector $\boldsymbol{p}_{t-k+1:t} \in \mathbb{R}^{7\times k}$ consist of the rotation axis, angle and transition vector:

$$\boldsymbol{x}_t = [\boldsymbol{s}_{t-k+1:t}; \boldsymbol{a}_{t-k+1:t}; \boldsymbol{p}_{t-k+1:t}] \tag{10}$$

$$\hat{\boldsymbol{z}}_j = \rho(\boldsymbol{x}_t) \tag{11}$$

The first part of ρ is a 2-layer MLP (256, 32) that map the input to a latent space. Then, a 3-layer 2D-CNN will extract the spatial-temporal information from the state-action-pose sequence $\boldsymbol{x}_t$. The input channel number, output channel number, kernel size, and stride of each layer are $[32, 32, 5, 5], [32, 32, 5, 1], [32, 32, 5, 1]$. Finally, there is a liner layer to flatten the output of ρ for estimating $\hat{\boldsymbol{z}}_t$.

Loss design: During training, we get access to the ground truth $\boldsymbol{e_j}$, and then obtain the ground truth $\boldsymbol{z_j}$ through encoder δ. So the regression loss is defined as:

$$\mathcal{L}_{\text{reg}} = \text{MSE}\left(\hat{\boldsymbol{z}}_j, \boldsymbol{z}_j\right) \tag{12}$$

However, the control performance cannot be guaranteed if we simply pursue the accurate prediction of calibration embedding. Following the classic adaptive control, we introduce the control-oriented loss to train the whole adaptive policy. Given the predicted calibration embedding$\hat{\boldsymbol{z}}_j$, we could predict the velocity command with the neural controller:

$${}^{T\bar{C}P}\hat{\mathbf{v}}^{\pi}_{TCP} = \pi(\boldsymbol{s}_t, \hat{\boldsymbol{z}}_j) \tag{13}$$

At the same time, we obtain the velocity supervision ${}^{T\bar{C}P}\mathbf{v}^{\text{GT}}_{TCP}$ as we discussed in Section III-B. So the velocity loss is defined as:

$$\begin{aligned}\mathcal{L}_{\text{dir}} &= 1 - \text{cosine\_similarity}({}^{T\bar{C}P}\hat{\mathbf{v}}^{\pi}_{TCP}, {}^{T\bar{C}P}\mathbf{v}^{\text{GT}}_{TCP}) \\ \mathcal{L}_{\text{norm}} &= \text{MSE}\left(l^{\text{pred}}, \mathcal{T}^{-1}\left(||{}^{T\bar{C}P}\mathbf{v}^{\text{GT}}_{TCP}||_2\right)\right)\end{aligned} \tag{14}$$

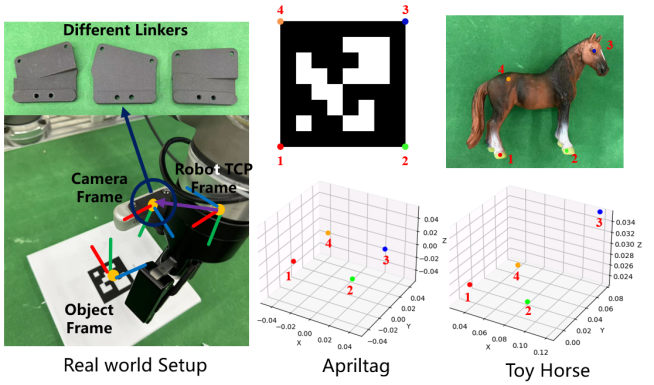

Fig. 3: We use the Apriltag and the Toy Horse for our target objects. Each of them have four feature points. We evaluate VS policy on real world robot with an uncalibrated camera can various linkers.

The final the control-oriented loss for adaptive policy training consists of both velocity loss and the regression loss:

$$\mathcal{L}_{\rho} = \mathcal{L}_{\text{reg}} + \mathcal{L}_{\text{dir}} + 0.5\mathcal{L}_{\text{norm}} \tag{15}$$

IV. System Settings

A. Simulation Settings

An environment including a virtual camera and the target object's 3D model is built in Pybullet. In each data collecting or model evaluation episode, the desired camera pose ${}^{o}\boldsymbol{T}_{c^*}$ is fixed at 15cm (AprilTag) or 25cm (Horse) above the target object a random initial camera pose ${}^{o}\boldsymbol{T}_{c}$ is sampled in the space 40cm above the target object with 0 to 10cm disturbance in XYZ translation. Both ${}^{o}\boldsymbol{T}_{c}$ and ${}^{o}\boldsymbol{T}_{c^*}$ ensure all keypoints within the camera's FoV.

When the camera is moving towards the desired pose, an episode is considered to be finished if the average error between the current 2D feature points and the desired feature points is smaller than a specified threshold:

$$\sum_{k=1}^{n} |u_k - u_k^*| + |v_k - v_k^*| \leq \delta_f \tag{16}$$

δ_f is set as 10 for simulation. Before reaching the desired pose, there are several situations that will trigger the early termination of the episode. These conditions are:

- The servo process exceeds $30s$ (every episode has a maximum steps of 300 with each step time of $0.1s$).
- The camera walks out the workspace.
- Any 2D feature point is out of the camera's FoV.

We use four criteria to analysis the servo performance: servo success rate (**SR**), servo time steps (**TS**), rotation error (**RE**) and translation error (**TE**). We calculate the transformation between the final TCP pose and the desired TCP pose ${}^{TCP^*}\mathbf{T}_{TCP}$.

Rotation Error (RE): The relative rotation ${}^{TCP^*}\mathbf{R}_{TCP}$ is converted into an axis-angle representation $\theta\mathbf{u}$. The rotation of camera is considered satisfactory if the deflection angle between the current pose and the desired pose is less than δ_r, *i.e.* $\theta < 5°$.

Translation Error (TE): The position of camera is considered to be in place if the displacement between the

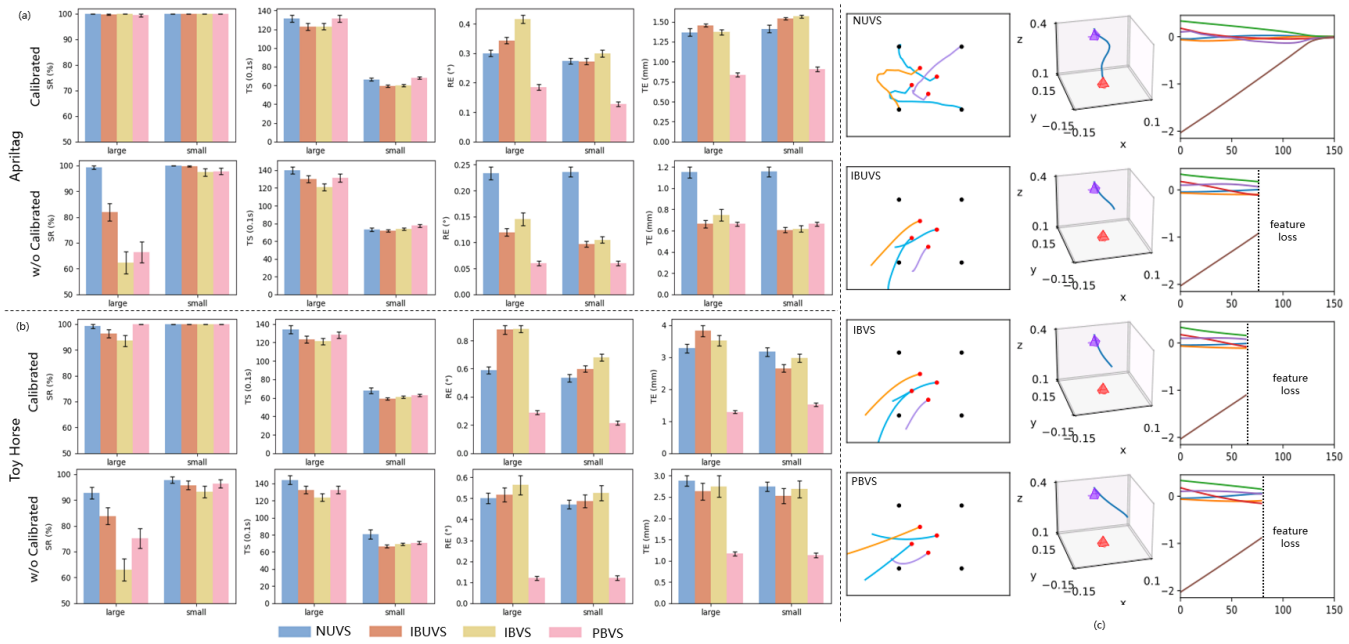

Fig. 4: Comparison in simulation to servo the (a) AprilTag and (b) Toy Horse with small or large initial offset given or without camera calibration. (c) shows the 2D, 3D and error curves of different methods given the same initial poses in uncalibrated scenarios.

TABLE I: Comparison with Kovis in simulation with large initial offset.

	Metrics	AprilTag		Horse	
		Ours	Kovis	Ours	Kovis
Calib.	SR(%)	**100**	**100**	**99.80**	98.20
	TS(0.1s)	**131.4±42.1**	145.4±52.7	**130.4±44.1**	140.8±49.5
	RE(°)	**0.31±0.10**	1.01±0.18	**0.50±0.12**	0.97±0.22
	TE(mm)	**1.38±0.30**	3.82±0.68	**2.67±0.68**	4.99±1.11
w/o Calib.	SR(%)	**99.20**	83.60	**91.8**	82.20
	TS(0.1s)	**139.5±45.1**	202.9±86.0	**142.7±55.7**	149.6±71.2
	RE(°)	**0.23±0.13**	1.19±0.31	**0.50±0.26**	1.07±0.45
	TE(mm)	**1.14±0.54**	5.12±1.62	**2.89±1.36**	5.40±2.09

current position and the desired position is less than δ_t, *i.e.* $\left\|^{TCP^*}\mathbf{t}_{TCP}\right\|_2^2 < 1cm$.

B. Real World Settings

The real world experiment is carried out on an UR5 robot which is shown in Fig. 3. Evaluation uses AprilTag as the target object as we can easily obtain its corners for feature correspondences. The settings of the real world experiment are same to the simulation except for δ_f is set as 20 and maximum steps is set as 300.

V. Experimental Results

In this section, we carried out a series of experiments to evaluate our method. Simulation experiments are performed on a computer with 16 Intel(R) Core(TM) i9-9900K 3.60GHz and one NVIDIA GeForce RTX 2080 SUPER. Real world experiments are performed on a computer with 12 Intel(R) Core(TM) i7-8700 3.20GHz and one NVIDIA GeForce GTX 1060. Our comparison methods are:

- **IBUVS**[12]: image based uncalibrated visual servo which use Kalman-Bucy filter to estimate Jacobian. For specific implementation, please refer to[11] or [12].
- **IBVS**[1]: classical image based visual servo implemented in [1].
- **PBVS**[1]: classical position based visual servo implemented in [1]. In practical use, the pose is estimated by PnP[41] given object's 3D and camera intrinsic.
- **Kovis**[3]: image based neural visual servo that uses the neural controller to predict velocity, which is differentiable and is trained on a fix camera calibration. We replace the Densenet based controller of Kovis to be the 4-layer multi-layer perception network (MLP) (256, 256, 128, 7) we used in base policy for fair comparison.

The goals of the experiments are:

- to compare the proposed NUVS with neural policy without calibration embedding modulation mechanism.
- to compare the proposed NUVS with classical visual servo policy and uncalibrated visual servo policy.
- to validate that the control-oriented traning is better than simply regression of calibration embedding.
- to evaluate all methods in real world when both calibration disturbance and observation error are introduced.

A. Simulation Evaluation

Experimental Setup: Simulation evaluations are performed on the Apriltag and the Toy Horse shown in Fig. 3. Metrics are calculated from 500 runs with random sampled initial poses and and calibration parameters. *Calibrated* means the calibration parameters are given to the methods while *w/o Calibrated* means the calibration parameters are not given to the methods. In *w/o Calibrated* cases, the same initial calibration parameters are given to all methods. Maximum initial pose offset between the initial and desired pose of *Large* offset is $\Delta\mathbf{r}_0 = ({}^{c^*}\mathbf{t}_c, \theta\mathbf{u})$: ${}^{c^*}\mathbf{t}_c = (10cm, 10cm, 25cm), \theta\mathbf{u} = (21.8°, 21.8°, 180°)$. Maximum initial pose offset between the initial and desired pose of *Small* offset is $\Delta\mathbf{r}_0 = ({}^{c^*}\mathbf{t}_c, \theta\mathbf{u})$: ${}^{c^*}\mathbf{t}_c = (6cm, 6cm, 15cm), \theta\mathbf{u} = (21.8°, 21.8°, 180°)$.

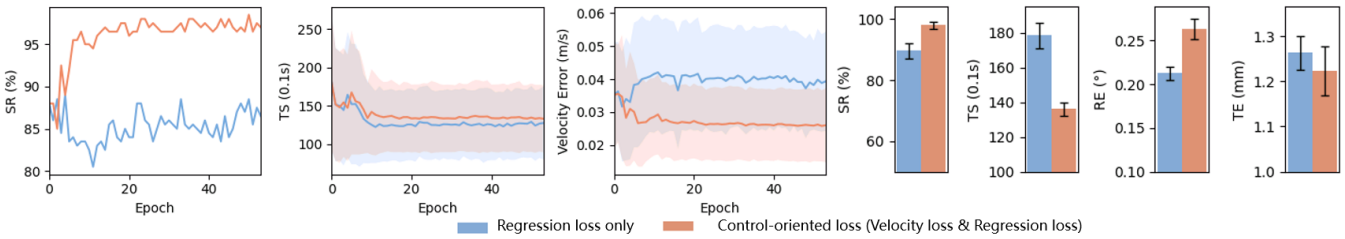

Fig. 5: The rad curve shows the training curve of NUVS supervised by the control-oriented loss which consists of both servo velocity loss and regression loss. The blue curve shows the training curve of NUVS supervised by only the regression loss. Velocity error shows the difference between the predicted control with the expert PBVS. The bar chart shows the performance comparison of the best models in 50 epochs training for each loss.

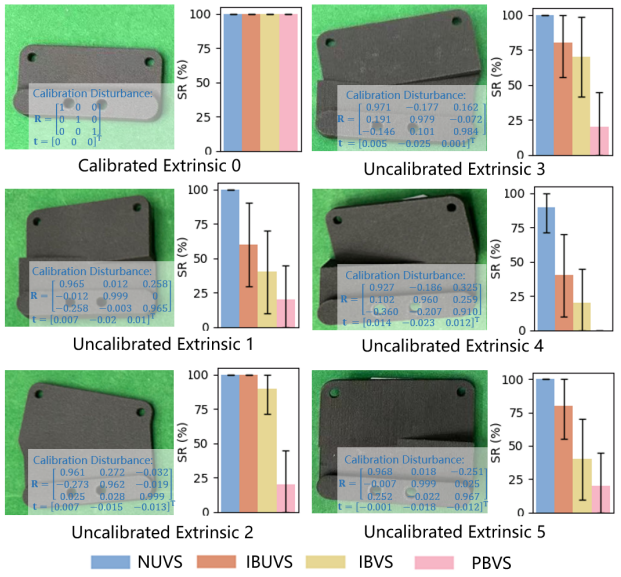

Fig. 6: Comparison of different methods in real world. For calibrated extrinsic 0, all of the methods achieve high success rate. Other linkers lead to various uncalibrated camera extrinsic and the camera intrinsic is also uncalibrated. NUVS outperforms other methods.

Comparison with Neural Policy: We compared NUVS vs. Kovis[3]. Kovis only take the feature correspondence as input, so it overfits to a fixed calibration parameter. It can be seen that in the calibrated scenario (the Calib. line in Table I), both NUVS and Kovis can achieve a relatively high servo success rate. When the calibration parameter is unknown (the w/o Calib. line in Table I), NUVS achieves a higher servo success rate because of having a calibration embedding adaption and a controller modulation mechanism.
Comparison with Classical Uncalibrated Visual Servo: As shown in Fig.4, in the calibrated scenario, all the methods achieves relatively high success rate. However, when servo the Toy Horse whose feature points are not in a plane in 3D space, IBVS and IBUVS fail in some of the cases due to the limited convergence basin especially when facing large initial offset. While PBVS still works, and NUVS which is supervised by PBVS preserves high success rate and precision as well.

When calibration disturbance are adding, the success rate of both IBVS and PBVS drops dramatically as these methods use erroneous calibration parameters to estimate Jacobian or pose. IBUVS online corrects the Jacobian estimation and reaches higer success rate than classical IBVS and PBVS. NUVS also demonstrates powerful adaption capabilities in these scenarios and have better performance than IBUVS.
Ablation of Control-oriented Training: Fig. 5 shows the performance comparison during training of NUVS's adaptive policy under different supervision. As we discussed in Section III-C, different from simply regression of the calibration embedding, control-oriented training uses both velocity loss and regression loss. Given the pretrained and freezed neural controller π, it enforces the adaption network ρ to predict a calibration embedding that makes π to better imitate the expert PBVS. The rad curve shows the performance of adaptive policy supervised by both servo velocity loss and regression loss. The blue curve shows the performance of adaptive policy supervised by only the regression loss. Control-oriented training can ensure a better servo performance. In another word, control-oriented training lets the adaptive policy better modulate the base policy.

B. Real World Evaluation

The real world setup is shown in Fig. 3. We use the Apriltag as the target object. We 3D print different linkers to link the camera and robot's TCP to get various camera extrinsic. The camera intrinsic is also uncalibrated. Moreover, observation error are introduced in real world. Fig. 6 shows the servo performance of NUVS, IBUVS, IBVS and PBVS under different camera extrinsic. It can be seen that PBVS has the worst adaptability to calibration disturbances, and the servo success rate drops significantly. IBVS has certain adaptability to disturbances, but the servo success rate has also declined. Since IBUVS can estimate the Jacobian online, it has better servo performance than IBVS. Our proposed NUVS can adapt well to different calibration disturbances and is better than all other methods.

VI. Conclusions

In this paper, we propose a neural uncalibrated visual servo policy for calibration disturbances adaption. The NUVS policy modulates a neural controller with calibration embedding and uses past observations to estimates calibration embedding in practical use. It is control-oriented supervised by the ideal PBVS, with extra regularization from latent embedding of calibration parameters. It bridges the disturbance adaption of classic VS methods, and the large convergence of learning based VS methods. Simulation and real world evaluations verify the effectiveness of NUVS, which achieves high success rate in visual servoing tasks with uncalibration disturbances.

REFERENCES

[1] F. Chaumette and S. Hutchinson, "Visual servo control. i. basic approaches," *IEEE Robotics Automation Magazine*, vol. 13, no. 4, pp. 82–90, 2006.

[2] Q. Bateux, E. Marchand, J. Leitner, F. Chaumette, and P. Corke, "Training deep neural networks for visual servoing," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 3307–3314.

[3] E. Y. Puang, K. P. Tee, and W. Jing, "Kovis: Keypoint-based visual servoing with zero-shot sim-to-real transfer for robotics manipulation," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 7527–7533.

[4] B. Thuilot, P. Martinet, L. Cordesses, and J. Gallice, "Position based visual servoing: keeping the object in the field of vision," in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, vol. 2, 2002, pp. 1624–1629 vol.2.

[5] D.-H. Park, J.-H. Kwon, and I.-J. Ha, "Novel position-based visual servoing approach to robust global stability under field-of-view constraint," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 12, pp. 4735–4752, 2012.

[6] W. J. Wilson, C. W. Hulls, and G. S. Bell, "Relative end-effector control using cartesian position based visual servoing," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 5, pp. 684–696, 1996.

[7] O. Kermorgant and F. Chaumette, "Combining ibvs and pbvs to ensure the visibility constraint," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011, pp. 2849–2854.

[8] A. Shademan and M. Jägersand, "Three-view uncalibrated visual servoing," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2010, pp. 6234–6239.

[9] J. Qian and J. Su, "Online estimation of image jacobian matrix by kalman-bucy filter for uncalibrated stereo vision feedback," in *Proceedings 2002 IEEE international conference on robotics and automation (cat. No. 02CH37292)*, vol. 1. IEEE, 2002, pp. 562–567.

[10] M. Bonkovic, A. Hace, and K. Jezernik, "Population-based uncalibrated visual servoing," *IEEE/ASME Transactions on Mechatronics*, vol. 13, no. 3, pp. 393–397, 2008.

[11] M. Hao and Z. Sun, "A universal state-space approach to uncalibrated model-free visual servoing," *IEEE/ASME Transactions on Mechatronics*, vol. 17, no. 5, pp. 833–846, 2011.

[12] J. Musić, M. Bonković, and M. Cecić, "Comparison of uncalibrated model-free visual servoing methods for small-amplitude movements: A simulation study," *International Journal of Advanced Robotic Systems*, vol. 11, no. 7, p. 108, 2014.

[13] F. Chaumette, "Potential problems of stability and convergence in image-based and position-based visual servoing," in *The confluence of vision and control*. Springer, 2007, pp. 66–78.

[14] H. Yu, A. Chen, K. Xu, Z. Zhou, W. Jing, Y. Wang, and R. Xiong, "A hyper-network based end-to-end visual servoing with arbitrary desired poses," *IEEE Robotics and Automation Letters*, 2023.

[15] S. M. Richards, N. Azizan, J.-J. Slotine, and M. Pavone, "Adaptive-control-oriented meta-learning for nonlinear systems," *arXiv preprint arXiv:2103.04490*, 2021.

[16] A. Kumar, Z. Fu, D. Pathak, and J. Malik, "Rma: Rapid motor adaptation for legged robots," *arXiv preprint arXiv:2107.04034*, 2021.

[17] J.-J. E. Slotine and W. Li, "On the adaptive control of robot manipulators," *The international journal of robotics research*, vol. 6, no. 3, pp. 49–59.

[18] F. Janabi-Sharifi, L. Deng, and W. J. Wilson, "Comparison of basic visual servoing methods," *IEEE/ASME Transactions on Mechatronics*, vol. 16, no. 5, pp. 967–983, 2011.

[19] G. Allibert, E. Courtial, and F. Chaumette, "Predictive control for constrained image-based visual servoing," *IEEE Transactions on Robotics*, vol. 26, no. 5, pp. 933–939, 2010.

[20] A. Ghasemi, P. Li, W.-F. Xie, and W. Tian, "Enhanced switch image-based visual servoing dealing with featuresloss," *Electronics*, vol. 8, no. 8, 2019.

[21] G. Allibert, E. Courtial, and F. Chaumette, "Predictive control for constrained image-based visual servoing," *IEEE Transactions on Robotics*, vol. 26, no. 5, pp. 933–939, 2010.

[22] A. Hajiloo, M. Keshmiri, W.-F. Xie, and T.-T. Wang, "Robust online model predictive control for a constrained image-based visual servoing," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 4, pp. 2242–2250, 2016.

[23] N. R. Gans and S. A. Hutchinson, "Stable visual servoing through hybrid switched-system control," *IEEE Transactions on Robotics*, vol. 23, no. 3, pp. 530–540, 2007.

[24] J. Qian and J. Su, "Online estimation of image jacobian matrix by kalman-bucy filter for uncalibrated stereo vision feedback," in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, vol. 1, 2002, pp. 562–567 vol.1.

[25] K. Hosoda and M. Asada, "Versatile visual servoing without knowledge of true jacobian," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'94)*, vol. 1, 1994, pp. 186–193 vol.1.

[26] M. Jagersand, O. Fuentes, and R. Nelson, "Experimental evaluation of uncalibrated visual servoing for precision manipulation," in *Proceedings of International Conference on Robotics and Automation*, vol. 4, 1997, pp. 2874–2880 vol.4.

[27] J. Armstrong Piepmeier, G. McMurray, and H. Lipkin, "A dynamic quasi-newton method for uncalibrated visual servoing," in *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*, vol. 2, 1999, pp. 1595–1600 vol.2.

[28] J. A. Piepmeier, G. V. McMurray, and H. Lipkin, "Uncalibrated dynamic visual servoing," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 1, pp. 143–147, 2004.

[29] J. Armstrong Piepmeier, B. Gumpert, and H. Lipkin, "Uncalibrated eye-in-hand visual servoing," in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, vol. 1, 2002, pp. 568–573 vol.1.

[30] M. Bonkovic, A. Hace, and K. Jezernik, "Population-based uncalibrated visual servoing," *IEEE/ASME Transactions on Mechatronics*, vol. 13, no. 3, pp. 393–397, 2008.

[31] A. Saxena, H. Pandya, G. Kumar, A. Gaud, and K. M. Krishna, "Exploring convolutional networks for end-to-end visual servoing," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 3817–3823.

[32] C. Yu, Z. Cai, H. Pham, and Q.-C. Pham, "Siamese convolutional neural network for sub-millimeter-accurate camera pose estimation and visual servoing," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 935–941.

[33] Y. Harish, H. Pandya, A. Gaud, S. Terupally, S. Shankar, and K. M. Krishna, "Dfvs: Deep flow guided scene agnostic image based visual servoing," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 9000–9006.

[34] P. Katara, Y. Harish, H. Pandya, A. Gupta, A. M. Sanchawala, G. Kumar, B. Bhowmick *et al.*, "Deepmpcvs: Deep model predictive control for visual servoing," *arXiv preprint arXiv:2105.00788*, 2021.

[35] N. Adrian, V.-T. Do, and Q.-C. Pham, "Dfbvs: Deep feature-based visual servo," *arXiv preprint arXiv:2201.08046*, 2022.

[36] B. Espiau, F. Chaumette, and P. Rives, "A new approach to visual servoing in robotics," *ieee Transactions on Robotics and Automation*, vol. 8, no. 3, pp. 313–326, 1992.

[37] R. Kelly, R. Carelli, O. Nasisi, B. Kuchen, and F. Reyes, "Stable visual servoing of camera-in-hand robotic systems," *IEEE/ASME transactions on mechatronics*, vol. 5, no. 1, pp. 39–48, 2000.

[38] S. Felton, E. Fromont, and E. Marchand, "Siame-se (3): regression in se (3) for end-to-end visual servoing," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 14 454–14 460.

[39] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[40] 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*. JMLR Workshop and Conference Proceedings, 2011, pp. 627–635.

[41] S. Peng, Y. Liu, Q. Huang, X. Zhou, and H. Bao, "Pvnet: Pixel-wise voting network for 6dof pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4561–4570.