Adapting Pretrained ViTs with Convolution Injector for Visuo-Motor Control

Dongyoon Hwang [*1] **Byungkun Lee** [*1] **Hojoon Lee** [1] **Hyunseung Kim** [1] **Jaegul Choo** [1]

Abstract

Vision Transformers (ViT), when paired with large-scale pretraining, have shown remarkable performance across various computer vision tasks, primarily due to their weak inductive bias. However, while such weak inductive bias aids in pretraining scalability, this may hinder the effective adaptation of ViTs for visuo-motor control tasks as a result of the absence of control-centric inductive biases. Such absent inductive biases include spatial locality and translation equivariance bias which convolutions naturally offer. To this end, we introduce **Co**nvolution **In**jector (**CoIn**), an add-on module that injects convolutions which are rich in locality and equivariance biases into a pretrained ViT for effective adaptation in visuo-motor control. We evaluate CoIn with three distinct types of pretrained ViTs (CLIP, MVP, VC-1) across 12 varied control tasks within three separate domains (Adroit, MetaWorld, DMC), and demonstrate that CoIn consistently enhances control task performance across all experimented environments and models, validating the effectiveness of providing pretrained ViTs with control-centric biases.[1,2]

1. Introduction

Developing intelligent robotic agents capable of precise visuo-motor control is an important area of research. A standard paradigm of developing such agents is to train the visual encoder and control policy end-to-end, using domain-specific control data (Levine et al., 2016). However, this approach limits the applicability of visuo-motor control policies in real-world scenarios due to the excessive amount of data required for learning and the lack of flexibility in adapting to new situations, such as unseen environments.

[*]Equal contribution [1]Kim Jaechul Graduate School of AI, KAIST. Correspondence to: Dongyoon Hwang <godnpeter@kaist.ac.kr>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

[1]Project page: https://godnpeter.github.io/CoIn
[2]Code: https://github.com/dojeon-ai/CoIn

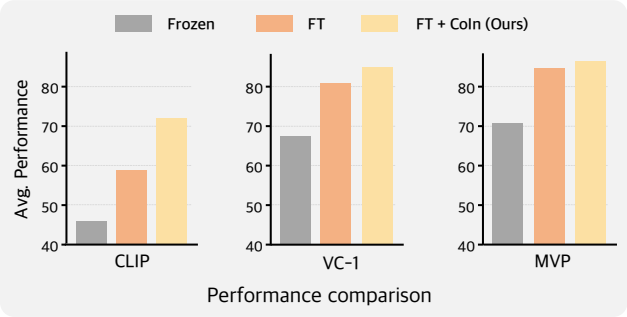


Figure 1. **Avg. performance across 12 visuo-motor control tasks.** Our model CoIn introduces convolutional inductive biases into ViTs, resulting in consistent performance improvements for various pretrained ViTs.

In the fields of computer vision and natural language processing, a large number of studies have shown that pretraining high-capacity models on large datasets demonstrate superior data efficiency and generalization capabilities compared to approaches trained from scratch (Dosovitskiy et al., 2021; Bommasani et al., 2021; Devlin et al., 2018; Brown et al., 2020). In response, for visuo-motor control, there has been a growing interest in utilizing large visual encoders pretrained on extensive and diverse datasets (Radosavovic et al., 2022; Hansen et al., 2021; Majumdar et al., 2023).

For visuo-motor control, Vision Transformers (ViT) (Dosovitskiy et al., 2021) emerges as an appealing choice as it achieved remarkable success in a wide range of computer vision tasks such as image classification (Dosovitskiy et al., 2021; Bao et al., 2022), object detection (Liu et al., 2021; Li et al., 2022) and semantic segmentation (Strudel et al., 2021; Kirillov et al., 2023). The success of ViTs is attributed to their weak inductive bias, which significantly enhances model performance when scaled with a large pretraining dataset and model size (Naseer et al., 2021; Yu et al., 2021; Mao et al., 2022; Chu et al., 2021; Dehghani et al., 2023).

Nonetheless, although the weak inductive bias of ViTs is advantageous for scaling during the pretraining phase, this characteristic may hinder their effective adaptation for visuo-motor control. For effective visuo-motor control, a visual encoder must (i) focus on the interaction area of interest, and (ii) track object and gripper positions with respect to

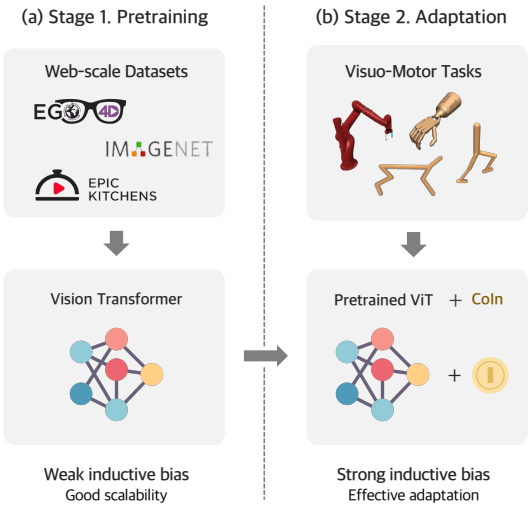


Figure 2. **Overall framework.** (Stage 1) The advent of open-sourced, large-scale ViTs pretrained with extensive web-scale datasets provides generalized, ready-to-go visual representations. (Stage 2) To adapt these pretrained ViTs for visuo-motor control, we finetune them with an additional light-weight module, CoIn, enhancing the ViT's ability to extract visual features beneficial for control, such as spatial locality and translation equivariance.

their changes in locations. ViTs inherently lack such properties due to their design. Such limitations of ViTs can be addressed by incorporating two specific biases that are naturally present in convolutional layers: (i) a bias towards spatial locality, and (ii) a bias for translation equivariance.

To this end, we introduce **Co**nvolution **In**jector (**CoIn**), a module designed to inject spatial locality and translation equivariance biases into a pretrained ViT for effective adaptation in visuo-motor control. CoIn is a simple and lightweight add-on module (3.6% of additional parameters to a standard ViT-B/16) designed to exploit the strengths of pretrained ViTs while providing advantageous inductive biases essential for visual control tasks. Specifically, CoIn extracts locality and translation equivariance-aware features through convolutional layers and integrates them into the ViT architecture using a cross-attention mechanism (see Figure 3). This integration enables the pretrained ViT to effectively leverage both its pretrained knowledge and newly obtained spatial prior features during adaptation for downstream control tasks. Therefore, CoIn eliminates the need to retrain pretrained ViTs from scratch with datasets and objectives specifically tailored for visual control applications.

To thoroughly evaluate the effectiveness of CoIn, we conduct extensive evaluations across 12 different visuo-motor control tasks within 3 distinct domains: Adroit (Rajeswaran et al., 2018), MetaWorld (Yu et al., 2020), and DMC (Tassa et al., 2018) for three different pretrained ViT visual encoders: CLIP (Radford et al., 2021), MVP (Radosavovic et al., 2022), and VC-1 (Majumdar et al., 2023). Our results demonstrate that CoIn consistently enhances downstream control task performance across all environments and with all pretrained ViTs. Notably, when paired with CLIP, finetuning with CoIn achieved a substantial 11.3 point increase in mean success over finetuning the baseline CLIP model. These findings suggest that the incorporation of locality and translation-equivariance-aware features plays a crucial role in enhancing the capabilities of ViTs for visuo-motor control tasks.

In summary, although ViTs gain advantages from large-scale pretraining due to their weak inductive bias, this same characteristic limits their adaptability for visuo-motor control tasks because of the absence of specific control-centric biases. Consequently, we introduce CoIn, a module which incorporates beneficial control-centric inductive biases, readily provided by convolutional layers, into large-scale pretrained ViTs. Our code is available at `https://godnpeter.github.io/CoIn`.

2. Related Work

2.1. Pretrained Visual Encoders for Control

Recently, pretraining effective visual encoders for control by leveraging large, diverse datasets from the internet has gain much interest from the research community (Parisi et al., 2022; Nair et al., 2022; Radosavovic et al., 2022; Majumdar et al., 2023; Wang et al., 2022; Yuan et al., 2022; Shah & Kumar, 2021). Specifically, PVR (Parisi et al., 2022) is among the initial investigations into the use of large pretrained visual encoders for control. It demonstrates that while doing behavior cloning, ResNet encoders (He et al., 2016) trained via self-supervised contrastive learning (He et al., 2020) can match the performance of state-based inputs. Further advancements are seen in R3M (Nair et al., 2022), which employs a temporal contrastive objective to learn representations for robotic control and VIP (Ma et al., 2022), which focuses on learning visual representations which reflect the distance between states and goals. Similarly, MVP (Radosavovic et al., 2022) and VC-1 (Majumdar et al., 2023) demonstrate the efficacy of ViTs pretrained with MAE (He et al., 2022) on extensive internet video and image data for robotic manipulations. As an alternative attempt, MOO (Stone et al., 2023) and RT-2 (Brohan et al., 2023) investigate the application of vision-language models pretrained on broad internet data, for improved robotic control and emergent reasoning. Unlike previous research which mainly focus on the performance of *frozen* weights in different control tasks, our work delves into the effectiveness and challenges of *finetuning* ViTs for control tasks, particularly in the context of imitation learning.

2.2. Integration of CNNs with Pretrained ViTs in Computer Vision

The integration of CNNs with pretrained ViTs to leverage their collective capabilities for various computer vision tasks has recently been investigated by the research community (Peng et al., 2021; Fang et al., 2023; Chen et al., 2022b; Ranftl et al., 2021; Hong et al., 2022). VitMatte (Yao et al., 2024) demonstrates the effectiveness of combining lightweight CNNs with a pretrained ViT for enhanced image matting. For dense prediction, DPT (Ranftl et al., 2021) introduces a randomly initialized CNN decoder, and ViT-Adapter (Chen et al., 2022b) utilizes a CNN-based adapter which embeds local semantic features into pretrained ViTs. MIMDET (Fang et al., 2023) employs a compact CNN encoder before the patch embedding layer of ViT, creating a CNN-ViT hybrid feature extractor for object detection.

Such hybrid models are particularly well-suited for visuo-motor control applications. This approach naturally integrates spatial locality and translation equivariance biases which lack in ViTs, but are essential for visuo-motor control. This eliminates the need for retraining with visuo-motor specific datasets for obtaining ViTs tailored towards visuo-motor control. However, despite active exploration in computer vision, the application of such convolutional bias integrated pretrained ViTs for visuo-motor control has been relatively unexplored (often relying on either standard ResNet or ViT models as previously mentioned in section 2.1). Our research aims to address this gap, exploring how pretrained ViTs can be effectively adapted for control tasks while fully leveraging their well-generalized features.

3. Method

Our objective is to enhance the capabilities of pretrained ViTs for visuo-motor control, by introducing control-centric inductive biases during the adaptation stage. To achieve this, we propose a simple yet effective add-on module, termed CoIn. Inspired by ViT-Adapter (Chen et al., 2022b), CoIn is composed of a lightweight CNN encoder and a cross-attention layer (Chen et al., 2021). This design enables the effective incorporation of locality and translation equivariant features extracted from the CNN encoder into ViT's patch embeddings for improved visuo-motor task performance. From here, we denote *convolutional inductive bias* to refer to both spatial locality and translation equivariance.

The ViT architecture is briefly described in Section 3.1, and the CNN module and cross-attention mechanism in CoIn are described in Section 3.2, 3.3, respectively.

3.1. Vision Transformer

In ViT, there are primarily two components: the patch embedding module and transformer encoder blocks (Dosovitskiy et al., 2021). For an image $X \in \mathbb{R}^{H \times W \times 3}$ (H, W denotes the image's resolution), the model segments the image into patches of size 16×16 through the patch embedding module. Then, these patches undergoes a three-step transformation: they are (1) flattened, (2) projected into D-dimensional vectors, and (3) augmented with positional embeddings. The resultant token (Z_0 in Eq. 1) are then fed sequentially through a series of transformer encoder blocks.

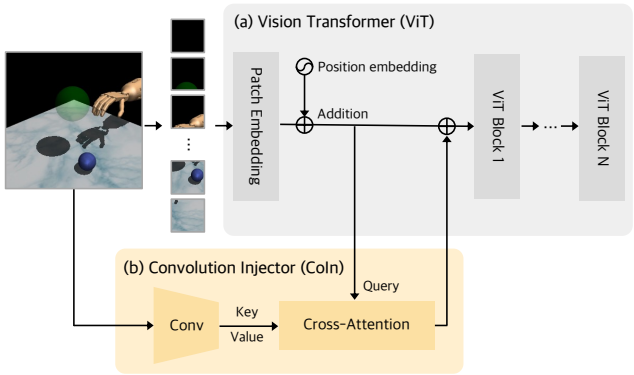


Figure 3. **Overall architecture of CoIn.** While leaving the (a) ViT architecture untouched, (b) CoIn incorporates two key modules: a CNN encoder, which captures spatial locality and translation equivariance rich features from the input image, and a cross attention module, which introduces such biases into the ViT patch token embeddings. Notably, these enhancements are seamlessly integrated without any modification to the overall ViT architecture.

$$
\begin{aligned}
Z_0 &= \text{PatchEmbedding}(X), & X &\in \mathbb{R}^{H \times W \times 3} \\
Z_l &= \text{Block}_l(Z_{l-1}), \quad l = 1...L, & Z_l &\in \mathbb{R}^{N \times D}
\end{aligned} \tag{1}
$$

L denotes the total number of transformer encoder blocks and N denotes the number of patches.

3.2. CNN Encoder

To adapt a standard pretrained ViT for control tasks, we introduce a lightweight CNN encoder (Figure 3b (left)). Its primary role is to generate features rich in spatial locality and translation equivariance bias which will later benefit the token embeddings Z_0 before they proceed through the transformer encoder blocks.

The design of the CNN encoder takes inspiration from the spatial prior module described in ViT-Adapter (Chen et al., 2022b). Initially, it uses a standard convolutional stem (He et al., 2016), which is followed by a series of stride-2 3×3 convolutions.

$$
\begin{aligned}
S &= \text{Stem}\ (X) & S &\in \mathbb{R}^{\frac{H}{4}\times\frac{W}{4}\times c_0} \\
\mathcal{F}_1 &= \text{Conv}_1\ (S) & \mathcal{F}_1 &\in \mathbb{R}^{\frac{H}{8}\times\frac{W}{8}\times c_1} \\
\mathcal{F}_2 &= \text{Conv}_2\ (\mathcal{F}_1) & \mathcal{F}_2 &\in \mathbb{R}^{\frac{H}{16}\times\frac{W}{16}\times c_2} \\
\mathcal{F}_3 &= \text{Conv}_3\ (\mathcal{F}_2) & \mathcal{F}_3 &\in \mathbb{R}^{\frac{H}{32}\times\frac{W}{32}\times c_3}
\end{aligned} \tag{2}
$$

where c_i denotes the hidden dimension of each layer.

As the output from the stem layer S passes through subsequent Conv_1 to Conv_3, we generate a feature pyramid consisting of multiple scales, represented as $\mathcal{F}_{\text{conv}} = \{\mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3\}$. Each scale of this pyramid corresponds to a different resolution, providing a comprehensive spatial representation of the input image. The inclusion of this multi-scale feature map array in our architecture enhances the model's ability to perceive spatial information at various resolutions, which is a key aspect for control tasks where recognizing different spatial scales is essential. Ablations on employing multi-scale feature is provided in Section 5.4.1.

Next, to makes these feature maps compatible with the ViT token embeddings, we apply 1×1 convolutions to each scale of the feature pyramid, which results in $\mathcal{F}_1 \in \mathbb{R}^{\frac{H}{8}\times\frac{W}{8}\times D}$, $\mathcal{F}_2 \in \mathbb{R}^{\frac{H}{16}\times\frac{W}{16}\times D}$, and $\mathcal{F}_3 \in \mathbb{R}^{\frac{H}{32}\times\frac{W}{32}\times D}$, where D matches the dimension of the ViT patch token embeddings Z_0.

3.3. Cross Attention Module

To incorporate the convolutional inductive bias rich features provided by the feature pyramid $\mathcal{F}_{\text{conv}}$ into a pretrained ViT, we utilize the multi-head cross attention mechanism (Vaswani et al., 2017; Alayrac et al., 2022).

Initially, each of the feature maps from $\mathcal{F}_{\text{conv}}$ is flattened and merged into a singular tensor $\mathcal{F}'_{\text{conv}} = \mathbb{R}^{(HW/8^2+HW/16^2+HW/32^2)\times D}$. Subsequently, we employ the output patch embeddings Z_0 from the ViT, as the query. $\mathcal{F}_{\text{conv}}$ is utilized both as the key and the value. This cross-attention mechanism enables the pretrained ViT to utilize the spatial locality and translation equivariance bias rich features extracted by the CNN encoder, which are important for downstream visuo-motor control tasks.

$$
\begin{aligned}
\hat{Z}_0 &= Z_0 + \text{CrossAttention}(Z_0, \mathcal{F}'_{\text{conv}}) \\
\hat{Z}_l &= \text{Block}_l(\hat{Z}_{l-1}), \quad l = 1 \ldots L
\end{aligned} \tag{3}
$$

The enriched outputs $\hat{Z}_0$, which are the sum of the outputs from the cross-attention module and the original Z_0, are then processed through the standard encoder blocks of the original ViT transformer.

This formulation ensures a seamless and effective integration of convolutional features into the ViT architecture. It allows for a feature representation enriched with spatial priors, while simultaneously leveraging the robust and powerful representations of a pretrained ViT.

3.4. Implementation and Computation Requirements

We note that CoIn exhibits a significantly lower computation footprint compared to a standard ViT, thereby minimizing the additional computational burden during the finetuning stage. To illustrate, while a standard ViT-B/16 (our primary experimental architecture) contains approximately 85.8M parameters, CoIn contains only approximately 3.1M parameters. This amounts to merely 3.6% of the parameter count of a ViT-B/16, highlighting CoIn's lightweight nature. Such a compact design makes CoIn an affordable add-on module to be additionally trained along with the ViT during the finetuning stage. Furthermore, in the interest of computational efficiency within the cross-attention layer, we adopt a linear sparse self-attention variant (Zhu et al., 2020; Chen et al., 2022b), which is recognized for its computational efficiency in terms of trainable parameters, training time, and memory compared to conventional global self-attention modules. Further implementation details are provided at Appendix C.

4. Experiment Setup

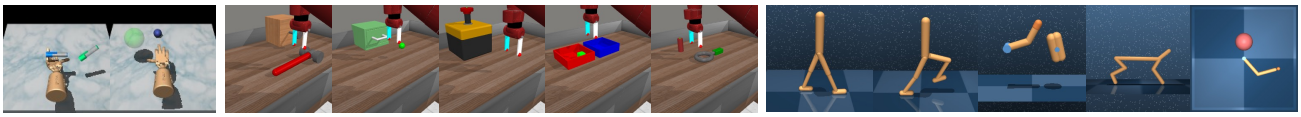
4.1. Environments

Here we describe the environments and tasks used in our evaluation. We consider a total of 12 tasks across three different domains: 2 tasks from Adroit (Rajeswaran et al., 2018), 5 tasks from MetaWorld (Yu et al., 2020), and 5 tasks from DMC (Tassa et al., 2018). We provide a brief description regarding the selected tasks below (See Figure 4).

Adroit (Rajeswaran et al., 2018) is a suite of tasks focused on dexterous manipulation. The agent is required to control a 28-DoF anthropomorphic hand to accomplish various goal-oriented activities in a virtual 3D environment. We focus on two challenging tasks from Adroit, 'Relocate' and 'Reorient-Pen', where the agent's objective is to either position an object at a specific target location or align it to a predetermined orientation. These tasks serve as a measure of the robot's precision and adaptability in complex manipulation tasks.

MetaWorld (Yu et al., 2020) requires an agent to control a Sawyer robot arm to perform various object manipulation tasks on a tabletop environment. Following prior work (Majumdar et al., 2023; Nair et al., 2022), we utilize five tasks from MetaWorld: Assembly, Bin Picking, Button Pressing, Drawer Opening, and Hammering.

Deepmind Control Suite (DMC) (Tassa et al., 2018) is a widely used benchmark for continuous control which involves low-level locomotion and manipulation of various difficulty. In our studies, we focus on five tasks from the



(a) Adroit | Pen, Relocate (b) MetaWorld | Hammer, Drawer, Button Press, Bin Picking, Assembly (c) DeepMind Control Suite | Walker, Finger Spin, Cheetah, Reacher

Figure 4. **Visualization of tasks used in our evaluation.** We utilize 2 tasks from Adroit, 5 tasks from Metaworld, and 5 tasks from DMC.

suite: Walker Stand, Walker Walk, Reacher Easy, Finger Spin, and Cheetah Run.

4.2. Models

In order to validate the efficacy of our approach, we experiment on three pretrained ViT encoders which have been widely utilized as a visual encoder for control tasks.

CLIP (Radford et al., 2021) is pretrained on a vast collection of web-scale image-text pairs, aligning image and language features effectively through contrastive learning. Its capabilities extend to a variety of tasks, including manipulation and navigation (Shridhar et al., 2022; Khandelwal et al., 2022). In line with existing work, our research explores its potential as a foundational visual encoder for both manipulation and locomotion tasks.

MVP (Radosavovic et al., 2022) focuses on spatial understanding by reconstructing randomly masked patches using a massive collection of Internet and egocentric data. MVP underlines the advantages of pretraining large visual encoders from web scale datasets for real-world robotic applications.

VC-1 (Majumdar et al., 2023) seeks to extend the achievements of MVP for pretrained visual representations in robotics. By coupling the ViT encoder and MAE pretraining objective on a more diverse dataset primarily composed of egocentric data, VC-1 attains competitive results in a wide array of visuo-motor control tasks.

4.3. Downstream Evaluation

In this paper, we focus on adapting pretrained visual representations for visuo-motor control tasks using behavior cloning (BC) with minimal expert trajectory data to effectively learn a control policy network $\pi(\cdot)$. The objective function is defined as:

$$L = \sum_{i=1}^{N} \sum_{t=1}^{H} ||a_t^i - \pi([z_t^i, p_t^i])||_2^2 \tag{4}$$

where a_t, z_t, and p_t denote the expert action, the encoded visual representation, and the proprioceptive information for trajectory i at timestep t, respectively.

Observations in the expert trajectory data consist of 256×256 RGB images, which are center-cropped to 224×224. For ViT models, the `[CLS]` token is used as the encoded visual observation feature input to the control policy network $\pi(\cdot)$, whereas for ResNet models, the final feature map after global average pooling serves as the encoded visual observation feature input. The default architecture utilized throughout all experiments is ViT-B/16 for ViT models and ResNet50 for ResNet models, unless otherwise specified.

For Adroit and MetaWorld tasks, agents receive proprioceptive data, which are concatenated to the encoded visual observation features before being fed into the control policy network $\pi(\cdot)$. In contrast, for DMC tasks, proprioceptive data is not available, so only the encoded visual observations are fed into $\pi(\cdot)$.

Following existing work (Hansen et al., 2022; Parisi et al., 2022; Majumdar et al., 2023; Nair et al., 2022), we utilize 100 expert demonstrations for Adroit and DMC, and 25 for MetaWorld, across a training span of 100 epochs. The visuo-motor control policy's performance is evaluated every 5 epochs, with the best success rate achieved during training reported across three independent runs for each task. For Adroit and MetaWorld, success rate serves as the primary metric, while normalized episode return is used for DMC. Further implementation details are provided in Appendix C.

5. Experiments

5.1. Main Results

E2E finetuning works. Previous studies have primarily evaluated the efficacy of pretrained visual representation for visuo-motor control tasks by freezing the visual encoders and finetuning only the control policy network, leaving end-to-end finetuning as an open question for future investigation (Parisi et al., 2022; Radosavovic et al., 2022; Nair et al., 2022; Ma et al., 2022). End-to-end finetuning within this domain hasn't always matched the success observed in other fields such as computer vision, occasionally resulting in suboptimal performance in visuo-motor control tasks. Previous research often suspect overfitting as a critical issue, suggesting the need of unique adaptation strategies such as performing further self-supervised pretraining on demonstration data to address these challenges (Yuan et al., 2022; Majumdar et al., 2023). Our findings, detailed in Appendix A, demonstrate that applying standard optimization strategies from the computer vision domain, such as weight decay and cosine learning rate scheduling (He et al., 2022),

Table 1. **Main results: CoIn with various pretrained ViTs.** Performance improvements achieved by incorporating CoIn across 12 tasks in three benchmarks (Adroit, MetaWorld, DMC) with three independent seeds. For each benchmark, we report the average performance and the average standard deviation of each task. CoIn is indicated in gray rows and the best results for each model are highlighted in bold. CoIn consistently improves performance for all models and across all benchmarks.

Backbone	Model	Training Strategy	Adroit	MetaWorld	DMC	Mean Success
ResNet50	VIP (Ma et al., 2022)	Frozen	58.0 ± 7.6	92.0 ± 3.5	64.4 ± 3.8	71.5
		Finetuned	63.3 ± 4.6	95.5 ± 3.9	82.4 ± 1.8	80.4
	R3M (Nair et al., 2022)	Frozen	61.3 ± 6.3	92.5 ± 2.9	69.8 ± 3.8	74.5
		Finetuned	78.7 ± 3.5	94.9 ± 3.5	81.8 ± 1.7	85.1
ViT-B	CLIP (Radford et al., 2021)	Frozen	38.7 ± 3.2	60.5 ± 5.1	37.4 ± 2.3	45.5
		Finetuned	47.3 ± 3.2	68.8 ± 8.1	62.8 ± 4.6	59.6
		Finetuned + CoIn	$\mathbf{52.7 \pm 6.2}$	$\mathbf{88.8 \pm 3.1}$	$\mathbf{71.1 \pm 3.7}$	**70.9** (+11.3)
	MVP (Radosavovic et al., 2022)	Frozen	58.0 ± 3.5	89.6 ± 5.0	64.6 ± 5.2	70.7
		Finetuned	82.0 ± 5.3	94.1 ± 4.9	77.4 ± 1.9	84.5
		Finetuned + CoIn	$\mathbf{83.3 \pm 4.6}$	$\mathbf{94.9 \pm 3.5}$	$\mathbf{80.5 \pm 2.5}$	**86.2** (+1.7)
	VC-1 (Majumdar et al., 2023)	Frozen	50.0 ± 5.4	86.7 ± 5.4	61.0 ± 3.2	65.9
		Finetuned	73.3 ± 5.2	93.9 ± 4.0	74.9 ± 3.5	80.7
		Finetuned+ CoIn	$\mathbf{77.3 \pm 5.1}$	$\mathbf{95.7 \pm 2.2}$	$\mathbf{80.7 \pm 4.2}$	**84.6** (+3.9)

significantly improves finetuning performance for large visual encoders in visuo-motor control, leading to enhanced task performance across all models and tasks (Table 1).

We hope this finding will encourage future research to explore and validate the adoption of computer vision finetuning practices and hyperparameters for thorough evaluation of pretrained visual encoders in visuo-motor control tasks.

Effectiveness of CoIn. The integration of CoIn with various pretrained ViTs leads to notable performance enhancements across all baseline ViT models and their associated control tasks, as detailed in Table 1. Specifically, when CoIn is combined with CLIP, there is a significant increase in the mean performance by 11.3 points. Furthermore, the addition of CoIn also benefits MVP and VC-1, boosting their mean performance by 1.7 and 3.9 points, respectively.

The unique efficacy of CoIn with CLIP, as compared to its integration with MVP and VC-1, can be ascribed to the distinct nature of CLIP's pretraining datasets. MVP and VC-1 are pretrained on datasets with an egocentric perspective, making them naturally compatible with environments such as Adroit and MetaWorld, which require egocentric visual inputs from robot agents. Conversely, CLIP, which is pretrained on diverse web-scale image-text pairs, does not initially possess these egocentric, control-centric features. By integrating CoIn, CLIP is endowed with control-oriented inductive biases, allowing for a significant enhancement in its ability to adapt features for motor control tasks. This demonstrates that CoIn can be especially beneficial for pretrained ViT models which lack control-centric visual features. We would also like to note that despite being pretrained with egocentric data, MVP and VC-1 still lack control-specific inductive biases necessary for certain tasks, as evidenced by the performance gains when incorporating CoIn.

Further comparison with ResNet-based pretrained visual representation methods reveals an intriguing aspect of CoIn's performance. Initially, R3M outperforms ViTs, indicating the advantage of inductive biases from convolution layers for these tasks. However, CoIn's integration significantly enhances ViT performance, allowing them to meet or even surpass R3M's success rates in cases like MVP. This demonstrates CoIn's effectiveness in adapting ViTs for control tasks, especially when the pretrained models lack control-oriented features. Full performance table is available in Appendix D.

5.2. Analysis of CoIn Visual Features

In this section, we perform an in-depth analysis regarding the distinct properties of the visual features learned by ViTs, both with and without the incorporation of CoIn. Our primary objective is to determine whether incorporating CoIn in a standard ViT does indeed induce convolutional inductive bias rich visual features. We utilize VC-1 as our baseline model throughout our analysis experiments in this section.

Capturing high frequency. CNNs excel in detecting detailed, high-frequency elements in images, including local details such as texture, edges and contours (Bai et al., 2022), which is an essential property for effective visuo-motor control. Thus, we assess whether CoIn helps ViTs in capturing such valuable high-frequency elements by examining the relative log amplitudes in the Fourier transformed feature maps (Chen et al., 2022b; Si et al., 2022; Park & Kim, 2022). Figure 5a illustrates our findings: finetuning ViTs with CoIn significantly improves its ability to detect high-frequency signals compared to a standard ViT. This highlights the cru-

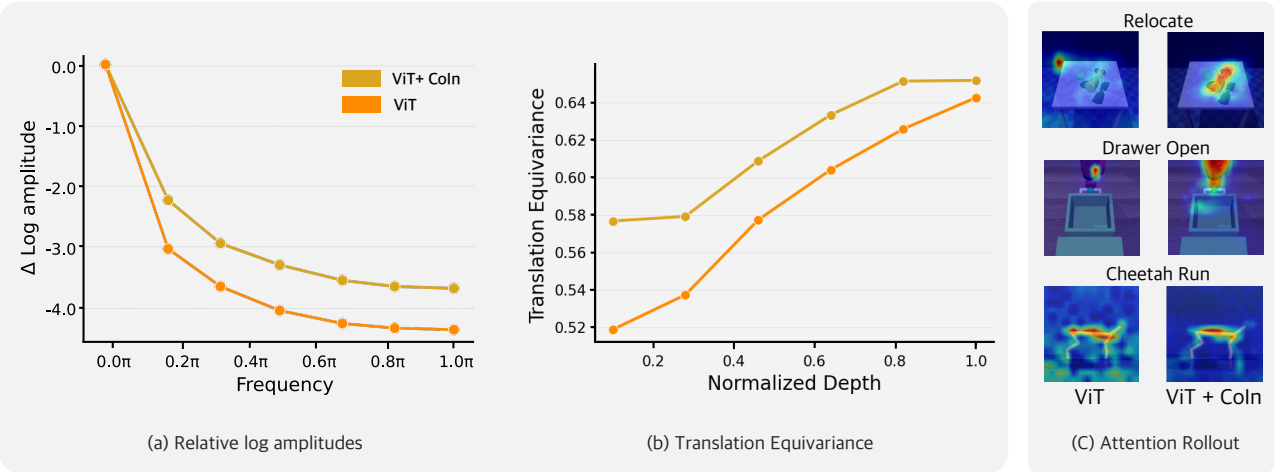


Figure 5. **(a) Comparison of the relative log amplitudes of Fourier-transformed feature maps.** ViT + CoIn incorporates beneficial inductive biases extracted from convolutional networks, allowing it to capture more high-frequency signals compared to ViT. **(b) Translation equivariance comparison.** ViT + CoIn enhances translation equivariance across intermediate representations within the ViT. **(c) Visualization of self-attention maps obtained through Attention Rollout.** ViT + CoIn exhibits improved focus on critical regions for visuo-motor control. All analysis were performed on VC-1 and averaged across all 12 tasks.

cial role of CoIn in effectively instilling spatial inductive biases to ViTs, thereby enhancing the performance of ViTs for downstream control tasks.

Translation equivariance. To evaluate whether the convolutional characteristics of CoIn enhance the learning of translation equivariant features for pretrained ViTs, we conducted a synthetic experiment following Bruintjes et al. (2023). Specifically, we computed the Pearson correlation between $f_{1:i}(T(X))$ and $T(f_{1:i}(X))$ for all $i = 1, 2, ..., N$, where T represents translations (diagonal shifts), f_i denotes the i-th intermediate layers of the ViT (i.e., $f_{1:i}(X) = f_i \cdot f_{i-1} \cdot ... \cdot f_1(x)$) and N refers to the total number of layers. Higher correlation values indicate that the model has learned higher translation equivariance.

While position embeddings in ViTs are known to present challenges for learning translation equivariance (Xu et al., 2023; Dai et al., 2021; Ding et al., 2023), CoIn alleviates this issue by directly injecting translation equivariance rich features into the output patches of the patch-embedding layer (Figure 3). As illustrated in Figure 5b, we observed that incorporating CoIn with ViTs enhances translation equivariance throughout the ViT intermediate representations.

Attention visualization. Additionally, we employ Attention Rollout (Abnar & Zuidema, 2020; Gildenblat, 2020) to visualize the self-attention maps of both VC-1 with and without CoIn (Figure 5c). This qualitative analysis further demonstrates that when equipped with CoIn, ViTs effectively focus more on image regions that are semantically relevant for visuo-motor control. This also highlights the efficacy in further providing ViTs with spatial locality and translation equivariance rich features via CoIn. More qualitative results can be found in Appendix G.

5.3. Comparison with Adapters

In this section, we aim to address a fundamental question: *Does the performance improvement of CoIn stem primarily from the utilization of additional parameters?* To assess this, we compare CoIn with two well established adapter-based methods, RoboAdapter (Sharma et al., 2023) and Adaptformer (Chen et al., 2022a). Although these adapter-based methods originally focus on parameter-efficient finetuning (PEFT), where the pretrained visual encoder is frozen and only the lightweight additional modules are finetuned for task adaptation (Houlsby et al., 2019; Hu et al., 2021), we explore a full finetuning variant of this approach where the pretrained visual encoder is finetuned alongside with the additional adapter modules. Such full finetuning variant provides an efficient means in incorporating additional task-specific parameters during finetuning.

Our findings, as detailed in Table 2, demonstrate that none of the adapter-based baseline methods match the performance of CoIn. Notably, for CLIP, only CoIn was able to enhance CLIP's performance while all other adapter baselines failed to provide any performance gains. This observation aligns with our explanation of why CoIn offers greater performance gains for CLIP compared to VC-1. The lack of performance gains from other adapter baselines can be attributed to their composition, which consists solely of MLPs and does not include any control-oriented inductive bias. As a result, these baselines are ineffective in helping CLIP learn

Table 2. **Full finetuning performance against adapter methods.** We report the mean performance of full finetuning across all 12 tasks in Adroit, MetaWorld, and DMC. Evaluations were conducted using CLIP and VC-1 with ViT-B. Underscored values indicate the hidden dimension size of the adapter modules. For detailed results, refer to Table 11.

Model	Module	# trainable params	Mean
CLIP	X	85.8M	59.6
	AdaptFormer_{64}	+1.2M	59.4
	RoboAdapter_{64}	+1.2M	59.2
	RoboAdapter_{192}	+3.5M	59.1
	CoIn	+3.1M	**70.9**
VC-1	X	85.8M	80.7
	AdaptFormer_{64}	+1.2M	82.3
	RoboAdapter_{64}	+1.2M	83.0
	RoboAdapter_{192}	+3.5M	82.5
	CoIn	+3.1M	**84.6**

control-oriented features. In contrast, CoIn's ability to impart control-oriented inductive biases significantly enhances CLIP's capacity to adapt features for motor control tasks.

This finding suggests that the effectiveness of CoIn is not merely due to the inclusion of additional trainable parameters. Instead, it significantly stems from the strategic integration of locality and translation equivariance biases, which are lacking in standard ViTs and are particularly beneficial for pretrained models lacking control-centric visual features.

Table 3. **PEFT performance against adapter methods.** We report the mean performance of parameter-efficient finetuning across all 12 tasks in Adroit, MetaWorld, and DMC. Evaluations were conducted using CLIP and VC-1 with ViT-B. Underscored values indicate the hidden dimension size of the adapter modules. For full results, refer to Table 12.

Model	Module	# trainable params	Mean
CLIP	X	–	45.5
	AdaptFormer_{64}	1.2M	51.3
	RoboAdapter_{64}	1.2M	50.1
	RoboAdapter_{192}	3.6M	50.3
	CoIn	3.3M	**67.9**
VC-1	X	–	65.9
	AdaptFormer_{64}	1.2M	78.8
	RoboAdapter_{64}	1.2M	78.8
	RoboAdapter_{192}	3.6M	78.9
	CoIn	3.3M	**79.5**

Additionally, we also conduct experiments under the PEFT scenario, which is the typical approach for training adapter-based methods. In this approach, the visual encoder is frozen and only the lightweight additional modules are finetuned. To achieve this, CoIn was minimally modified to include additional lightweight bottleneck MLP layers (merely 0.2M additional parameters) for the first two ViT encoder block. These layers are designed to process novel patch embeddings enriched with convolutional inductive biases, which the frozen pretrained ViT had not previously encountered.

Despite CoIn not being originally designed for parameter-efficient transfer, the results in Table 3 demonstrate CoIn's superior performance against traditional adapter methods. This highlights the potential of our approach for parameter efficient transfer learning of visual encoders in control tasks. In addition, similar to the full finetuning scenario, the apparent effectiveness of CoIn over other adapter baselines for CLIP further emphasizes the effectiveness of CoIn's ability to inject control-centric inductive biases crucial for visuo-motor tasks into pretrained ViTs which lack such biases,

5.4. Ablation Study

5.4.1. Feature Pyramid Components

In Section 3.2, we discussed the rationale behind incorporating multi-scale feature maps in CoIn instead of relying solely on a single feature map from its convolutional module. The central idea is that multi-scale feature maps can significantly improve the representational capabilities of visuo-motor control policies by capturing objects at different scales. This multi-scale approach is deemed critical for visuo-motor tasks, where the size and appearance of objects or obstacles can greatly vary. To evaluate our hypothesis, we evaluated a variant of CoIn that exclusively uses the $\mathcal{F}_2$ feature map. This feature map matches the resolution of ViT-B/16, meaning it is $\frac{1}{16}$th the size of the input resolution and does not offer a variety of feature scales.

Table 4. **Feature Pyramid Comparison.** Compared to using a single-scale feature map $\{\mathcal{F}_2\}$, utilizing multi-scale feature map $\{\mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3\}$ is beneficial for CoIn.

Model & Strategy	Feature Pyramid	Adroit	Meta-World	DMC	Mean
CLIP + Finetuned	X	47.3 ± 3.2	68.8 ± 8.1	62.8 ± 4.6	59.6
	$\{\mathcal{F}_2\}$	51.3 ± 5.2	86.1 ± 4.0	73.9 ± 3.3	70.4
	$\{\mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3\}$	52.7 ± 6.2	88.8 ± 3.1	71.1 ± 3.7	**70.9**
VC-1 + Finetuned	X	73.3 ± 5.2	93.9 ± 4.0	74.9 ± 3.5	80.7
	$\{\mathcal{F}_2\}$	76.0 ± 3.5	95.2 ± 3.4	79.9 ± 3.1	83.7
	$\{\mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3\}$	77.3 ± 5.1	95.7 ± 2.2	80.7 ± 4.2	**84.6**

Table 4 indicates that even the single-scale variants significantly improves the performance against the vanilla baseline, clearly highlighting the benefits of integrating inductive biases tailored to control tasks. Additionally, incorporating multi-scale feature maps yielded even better results, demonstrating the efficacy of providing varied scale perspectives.

We note that the integration of multi-scale feature maps into CoIn is both efficient and cost-effective, as they are intrinsically generated from the lightweight convolutional layers. Given these empirical observations, we set the multi-scale feature pyramid as our standard setup for CoIn.

5.4.2. Scaling Study

To further understand the effectiveness of CoIn on larger ViT models, we performed experiments where CoIn is applied to ViT-L/14 across all 12 tasks considered in this work. Table 5 details the results of finetuning VC-1 with and without CoIn, for both ViT-B/16 and ViT-L/14. For performance results of each benchmark, refer to Table 13.

Table 5. **CoIn with larger scale.** CoIn boosts visuo-motor control task performance for both scales, where ViT-B + CoIn outperforms ViT-L with significantly fewer parameters.

Model & Strategy	Backbone Scale & Additional Module	# trainable params	Mean
VC-1+ Finetuned	ViT-B	85.8M	80.7
	ViT-B + CoIn	88.9M	84.6
	ViT-L	303.3M	83.4
	ViT-L + CoIn	307.7M	84.5

The findings in Table 5 lay out three important aspects. First, finetuning larger pretrained encoders, as expected, yield better performance than smaller pretrained encoders in downstream visuo-motor control tasks. Second, the benefits of injecting convolutional inductive bias via CoIn works in tandem with increasing model size, enhancing performance regardless of the model's scale. Lastly, a particular surprising observation is that when ViT-B is paired with CoIn, despite having significantly fewer parameters, it outperforms ViT-L. This emphasizes the role of CoIn in enhancing smaller encoders to reach the efficacy levels of their larger counterparts.

Overall, these findings underscore the scalability and efficacy of CoIn in enhancing ViT models for complex visuo-motor control tasks, revealing its potential for application in larger encoders.

6. Discussion and Conclusion

In our study, we explore the challenges encountered by pretrained ViTs when applied to visuo-motor tasks. Particularly, while the weak inductive bias of ViTs are advantageous for large-scale pretraining, it limits their applicability in control-specific scenarios. To address this, we introduce a simple lightweight module, CoIn, designed to inject ViTs with convolutional inductive biases which are beneficial in performing effective visuo-control. This allows pretrained ViTs to leverage both their strong visual representations and beneficial biases for downstream visuo-motor control tasks provided by CoIn. Our thorough evaluation across a variety of visuo-motor control tasks confirms the consistent advantages and efficiency of CoIn.

In this work, we primarily focus on learning policies using behavior cloning, to highlight how CoIn effectively enhances ViTs for control tasks under limited data availability. We believe that extending CoIn to reinforcement learning for complex robotic tasks is a valuable avenue for future investigation. Moreover, while our current experiments are conducted within simulated environments, real-world robot experiments may present additional challenges and leave the evaluation of CoIn on real-world hardware as future work.

Acknowledgements

This work was supported by Institute for Information communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No.RS-2019-II190075 Artificial Intelligence Graduate School Program(KAIST), RS-2021-II212068 Artificial Intelligence Innovation Hub) and the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. NRF-2022R1A2B5B02001913).

Impact Statement

Our add-on module, CoIn, which injects convolutional inductive biases into pretrained Vision Transformers (ViTs), enables efficient adaptation of pretrained ViTs for visuo-motor control tasks. We anticipate that as foundation models such as ViTs continue to advance in the computer vision field, the performance of visuo-motor control systems will improve in tandem. Moreover, we acknowledge the potential risks associated with the current rapid advancements and potential misuse of such technologies. However, we believe that there are no specific ethical considerations in this paper which we feel must be specifically highlighted here.

References

Abnar, S. and Zuidema, W. Quantifying attention flow in transformers. *Proc. the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2020.

Alayrac, J.-B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., et al. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35:23716–23736, 2022.

Bai, J., Yuan, L., Xia, S.-T., Yan, S., Li, Z., and Liu, W. Improving vision transformers by revisiting high-frequency components. In *Proc. of the European Conference on Computer Vision (ECCV)*, pp. 1–18. Springer, 2022.

Bao, H., Dong, L., Piao, S., and Wei, F. Beit: Bert pre-training of image transformers. *International Conference on Learning Representations*, 2022.

Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora,

S., von Arx, S., Bernstein, M. S., Bohg, J., Bosselut, A., Brunskill, E., et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

Brohan, A., Brown, N., Carbajal, J., Chebotar, Y., Chen, X., Choromanski, K., Ding, T., Driess, D., Dubey, A., Finn, C., et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901, 2020.

Bruintjes, R.-J., Motyka, T., and van Gemert, J. What affects learned equivariance in deep image recognition models? 2023.

Cai, Z. and Vasconcelos, N. Cascade r-cnn: High quality object detection and instance segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 43 (5):1483–1498, 2019.

Chen, C.-F. R., Fan, Q., and Panda, R. Crossvit: Cross-attention multi-scale vision transformer for image classification. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 357–366, 2021.

Chen, S., Ge, C., Tong, Z., Wang, J., Song, Y., Wang, J., and Luo, P. Adaptformer: Adapting vision transformers for scalable visual recognition. *Advances in Neural Information Processing Systems*, 35:16664–16678, 2022a.

Chen, X. and He, K. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 15750–15758, 2021.

Chen, Z., Duan, Y., Wang, W., He, J., Lu, T., Dai, J., and Qiao, Y. Vision transformer adapter for dense predictions. In *The Eleventh International Conference on Learning Representations*, 2022b.

Chu, X., Tian, Z., Wang, Y., Zhang, B., Ren, H., Wei, X., Xia, H., and Shen, C. Twins: Revisiting the design of spatial attention in vision transformers. *Advances in Neural Information Processing Systems*, 34:9355–9366, 2021.

Clark, K., Luong, M.-T., Le, Q. V., and Manning, C. D. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*, 2020.

Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., and Wei, Y. Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pp. 764–773, 2017.

Dai, Z., Liu, H., Le, Q. V., and Tan, M. Coatnet: Marrying convolution and attention for all data sizes. *Proc. the Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Dehghani, M., Djolonga, J., Mustafa, B., Padlewski, P., Heek, J., Gilmer, J., Steiner, A. P., Caron, M., Geirhos, R., Alabdulmohsin, I., et al. Scaling vision transformers to 22 billion parameters. In *Proc. the International Conference on Machine Learning (ICML)*, 2023.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Ding, P., Soselia, D., Armstrong, T., Su, J., and Huang, F. Reviving shift equivariance in vision transformers. *arXiv preprint arXiv:2306.07470*, 2023.

Dong, X., Bao, J., Zhang, T., Chen, D., Gu, S., Zhang, W., Yuan, L., Chen, D., Wen, F., and Yu, N. Clip itself is a strong fine-tuner: Achieving 85.7% and 88.0% top-1 accuracy with vit-b and vit-l on imagenet. *arXiv preprint arXiv:2212.06138*, 2022.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. An image is worth 16x16 words: Transformers for image recognition at scale. *International Conference on Learning Representations*, 2021.

Fang, Y., Yang, S., Wang, S., Ge, Y., Shan, Y., and Wang, X. Unleashing vanilla vision transformer with masked image modeling for object detection. *Proc. of the IEEE international conference on computer vision (ICCV)*, 2023.

Gildenblat, J. Exploring explainability for vision transformers, 2020. URL https://github.com/jacobgil/vit-explain. Accessed: 2023-01-14.

Hansen, N., Su, H., and Wang, X. Stabilizing deep q-learning with convnets and vision transformers under data augmentation. In *Advances in Neural Information Processing Systems*, 2021.

Hansen, N., Yuan, Z., Ze, Y., Mu, T., Rajeswaran, A., Su, H., Xu, H., and Wang, X. On pre-training for visuomotor control: Revisiting a learning-from-scratch baseline. *arXiv preprint arXiv:2212.05749*, 2022.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

He, K., Gkioxari, G., Dollar, P., and Girshick, R. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9729–9738, 2020.

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick, R. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16000–16009, 2022.

Hong, Y., Pan, H., Sun, W., Yu, X., and Gao, H. Representation separation for semantic segmentation with vision transformers. *ArXiv Preprint*, 2022.

Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., Attariyan, M., and Gelly, S. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pp. 2790–2799. PMLR, 2019.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Khandelwal, A., Weihs, L., Mottaghi, R., and Kembhavi, A. Simple but effective: Clip embeddings for embodied ai. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14829–14838, 2022.

Kirillov, A., He, K., Girshick, R., Rother, C., and Dollar, P. Panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W.-Y., et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023.

Levine, S., Finn, C., Darrell, T., and Abbeel, P. End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research*, 17(1):1334–1373, 2016.

Li, Y., Mao, H., Girshick, R., and He, K. Exploring plain vision transformer backbones for object detection. In *European Conference on Computer Vision*, pp. 280–296. Springer, 2022.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10012–10022, 2021.

Ma, Y. J., Sodhani, S., Jayaraman, D., Bastani, O., Kumar, V., and Zhang, A. Vip: Towards universal visual reward and representation via value-implicit pre-training. *arXiv preprint arXiv:2210.00030*, 2022.

Majumdar, A., Yadav, K., Arnaud, S., Ma, Y. J., Chen, C., Silwal, S., Jain, A., Berges, V.-P., Wu, T., Vakil, J., Abbeel, P., Malik, J., Batra, D., Lin, Y., Maksymets, O., Rajeswaran, A., and Meier, F. Where are we in the search for an artificial visual cortex for embodied intelligence? In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Mao, X., Qi, G., Chen, Y., Li, X., Duan, R., Ye, S., He, Y., and Xue, H. Towards robust vision transformer. In *Proc. of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2022.

Nair, S., Rajeswaran, A., Kumar, V., Finn, C., and Gupta, A. R3m: A universal visual representation for robot manipulation. *Conference on Robot Learning*, 2022.

Naseer, M. M., Ranasinghe, K., Khan, S. H., Hayat, M., Shahbaz Khan, F., and Yang, M.-H. Intriguing properties of vision transformers. *Proc. the Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Parisi, S., Rajeswaran, A., Purushwalkam, S., and Gupta, A. The unsurprising effectiveness of pre-trained vision models for control. In *International Conference on Machine Learning*, 2022.

Park, N. and Kim, S. How do vision transformers work? In *International Conference on Learning Representations*, 2022.

Peng, Z., Huang, W., Gu, S., Xie, L., Wang, Y., Jiao, J., and Ye, Q. Conformer: Local features coupling global representations for visual recognition. In *Proc. of the IEEE international conference on computer vision (ICCV)*, 2021.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.

Radosavovic, I., Xiao, T., James, S., Abbeel, P., Malik, J., and Darrell, T. Real-world robot learning with masked visual pre-training. In *6th Annual Conference on Robot Learning*, 2022. URL https://openreview.net/forum?id=KWCZfuqshd.

Rajeswaran, A., Kumar, V., Gupta, A., Vezzani, G., Schulman, J., Todorov, E., and Levine, S. Learning Complex Dexterous Manipulation with Deep Reinforcement Learning and Demonstrations. In *Proceedings of Robotics: Science and Systems (RSS)*, 2018.

Ranftl, R., Bochkovskiy, A., and Koltun, V. Vision transformers for dense prediction. *Proc. of the IEEE international conference on computer vision (ICCV)*, 2021.

Shah, R. M. and Kumar, V. Rrl: Resnet as representation for reinforcement learning. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 9465–9476. PMLR, 18–24 Jul 2021.

Shang, W., Wang, X., Srinivas, A., Rajeswaran, A., Gao, Y., Abbeel, P., and Laskin, M. Reinforcement learning with latent flow. *Advances in Neural Information Processing Systems*, 34:22171–22183, 2021.

Sharma, M., Fantacci, C., Zhou, Y., Koppula, S., Heess, N., Scholz, J., and Aytar, Y. Lossless adaptation of pretrained vision models for robotic manipulation. *arXiv preprint arXiv:2304.06600*, 2023.

Shridhar, M., Manuelli, L., and Fox, D. Cliport: What and where pathways for robotic manipulation. In *Conference on Robot Learning*, pp. 894–906. PMLR, 2022.

Si, C., Yu, W., Zhou, P., Zhou, Y., Wang, X., and Yan, S. Inception transformer. *Proc. the Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Sovrasov, V. ptflops: a flops counting tool for neural networks in pytorch framework, 2024. URL https://github.com/sovrasov/flops-counter.pytorch.

Steiner, A., Kolesnikov, A., Zhai, X., Wightman, R., Uszkoreit, J., and Beyer, L. How to train your vit? data, augmentation, and regularization in vision transformers. *arXiv preprint arXiv:2106.10270*, 2021.

Stone, A., Xiao, T., Lu, Y., Gopalakrishnan, K., Lee, K.-H., Vuong, Q., Wohlhart, P., Zitkovich, B., Xia, F., Finn, C., et al. Open-world object manipulation using pre-trained vision-language models. *arXiv preprint arXiv:2303.00905*, 2023.

Strudel, R., Garcia, R., Laptev, I., and Schmid, C. Segmenter: Transformer for semantic segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 7262–7272, 2021.

Tassa, Y., Doron, Y., Muldal, A., Erez, T., Li, Y., de Las Casas, D., Budden, D., Abdolmaleki, A., Merel, J., Lefrancq, A., Lillicrap, T. P., and Riedmiller, M. A. Deepmind control suite. *CoRR*, abs/1801.00690, 2018.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Wang, C., Luo, X., Ross, K., and Li, D. Vrl3: A data-driven framework for visual deep reinforcement learning. *Advances in Neural Information Processing Systems*, 35: 32974–32988, 2022.

Xu, R., Yang, K., Liu, K., and He, F. $e(2)$-equivariant vision transformer. In *Proc. the Conference on Uncertainty in Artificial Intelligence (UAI)*, 2023.

Yao, J., Wang, X., Yang, S., and Wang, B. Vitmatte: Boosting image matting with pre-trained plain vision transformers. *Information Fusion*, 103:102091, 2024. ISSN 1566-2535. doi: https://doi.org/10.1016/j.inffus.2023.102091. URL https://www.sciencedirect.com/science/article/pii/S1566253523004074.

Yu, T., Quillen, D., He, Z., Julian, R., Hausman, K., Finn, C., and Levine, S. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on robot learning*, pp. 1094–1100. PMLR, 2020.

Yu, T., Li, X., Cai, Y., Sun, M., and Li, P. Rethinking token-mixing mlp for mlp-based vision backbone. *Proc. British Machine Vision Conference (BMVC)*, 2021.

Yuan, Z., Xue, Z., Yuan, B., Wang, X., Wu, Y., Gao, Y., and Xu, H. Pre-trained image encoder for generalizable visual reinforcement learning. In *Advances in Neural Information Processing Systems*, 2022.

Zhu, X., Su, W., Lu, L., Li, B., Wang, X., and Dai, J. Deformable detr: Deformable transformers for end-to-end object detection. In *International Conference on Learning Representations*, 2020.

A. Finetuning strategy

Prior research mainly evaluate pretrained visual encoder for visuo-motor control tasks by keeping them frozen and finetuning only the control policy. However, we argue that this approach does not fully assess the capabilities of pretrained visual encoders for visuo-motor control tasks. To understand their effectiveness, it is crucial to examine the effectiveness of pretrained visual encoders both when they are frozen and when they are finetuned. For instance, previous studies have suggested that linear probing may not accurately correlate with transfer learning performance (Chen & He, 2021), with some findings showing inconsistent rankings across tasks (See Figure 9 in (He et al., 2022))

Despite this, how to optimize pretrained visual encoder for visuo-motor control tasks is an under-researched question which has not received the attention it should by the research community. Addressing this gap, our experiments demonstrate that applying ViT finetuning strategies commonly utilized by the computer vision community are also effective for visuo-motor control tasks.

Table 6. **Finetuning strategy ablation**. Here we present the mean succes of VC-1 with ViT-B finetuned across all 12 tasks in Adroit, Metaworld, and DMC. The grey row indicates our default setup for finetuning ViTs. Cosine LR indicates cosine learning rate decay and LLDR indicates layer-wise learning rate decay.

	Finetune	Weight decay	Cosine LR	LLDR	Mean Success ↑
(a)	-	-	-	-	65.9
(b)	✓	-	-	-	55.1
(c)	✓	✓	-	-	58.9
(d)	✓	✓	✓	-	76.5
(e)	✓	✓	✓	✓	80.7

Our results, as shown in Table 6 (a) and (b), first reveal an initially counter-intuitive finding which has often been observed by previous research (Yuan et al., 2022; Majumdar et al., 2023): simply finetuning VC-1 with ViT-B results in deteriorated performance compared to its frozen counterpart. This is counter intuitive since further training the visual encoder on in-domain specific data should increase performance, not deteriorate the performance of the control policy. We speculate that the main cause of this performance deterioration is due to overfitting. By implementing a combination of weight decay, cosine learning rate decay, and layer-wise learning rate decay (Bao et al., 2022; Clark et al., 2020), which are techniques commonly used in finetuning ViTs (He et al., 2022; Bao et al., 2022; Dong et al., 2022; Steiner et al., 2021), we observe significant performance improvements. Specifically, applying weight decay led to a 3.8 points increase ($55.1 \rightarrow 58.9$), cosine learning rate decay resulted in a further 17.6 point boost ($58.9 \rightarrow 76.5$), and layer-wise learning rate decay (Clark et al., 2020; Bao et al., 2022) added an additional performance improvement of 4.2 points ($76.5 \rightarrow 80.7$). These strategies demonstrate the potential of finetuning to unlock the full capabilities of pretrained visual encoders for visuo-motor control tasks.

B. Comparison with ViT-Adapter

Here, we describe in detail how CoIn differs compared to the context of existing computer vision literature which introduce convolutional inductive biases to the ViT architecture, particularly in relation to ViT-Adapter (Chen et al., 2022b). We provide a detailed explanation to elucidate the difference between CoIn and ViT-Adapter and illustrate the advancements that CoIn offers through experimental results.

We would first like to clarify that CoIn and ViT-Adapter differ in (i) motivation and (ii) practical implementation. The main motivation behind CoIn is in *injecting convolutional inductive biases into pretrained ViTs*, as spatial locality and translation equivariance are beneficial properties in performing precise visuo-motor control. In contrast, the principal motivation behind ViT-Adapter is to construct an effective multi-scale feature map from pretrained ViT representations. This multi-scale feature map is subsequently used by dense prediction heads (He et al., 2017; Cai & Vasconcelos, 2019; Kirillov et al., 2019) to perform tasks such as object detection and semantic segmentation. Although CoIn incorporates elements inspired by ViT-Adapter, this difference in their motivations results in a key distinction: CoIn does not require the Extractor module present in ViT-Adapter and operates exclusively with the Injector module.

In ViT-Adapter, the Extractor module constructs effective multi-scale convolutional feature maps $\mathcal{F}_{\text{conv}} = \{\mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3\}$ by distilling representations from a pretrained ViT. This feature map serves as the output of ViT-Adapter. Conversely, CoIn focuses on injecting convolutional inductive bias rich features into pretrained ViTs for effective visuo-motor control.

Consequently, the Injector module is essential for CoIn while the Extractor module is unnecessary from a motivation standpoint. In addition, CoIn employs the ViT `[CLS]` token, rather than the multi-scale feature maps $\mathcal{F}_{\text{conv}} = \{\mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3\}$ for predicting actions. This further underscores the redundancy of the Extractor module in terms of practical implementation. Therefore, driven by its core motivation and implementation strategy, CoIn solely utilizes the Injector module.

Table 7. **Performance comparison between CoIn and ViT-Adapter**. We report the mean performance results of CoIn against ViT-Adapter across all 12 tasks in three benchmarks (Adroit, MetaWorld, DMC) with three independent seeds. CoIn significantly outperforms ViT-Adapter in terms of computational cost, inference speed, and mean performance.

Model (ViT-B/16)	Additional component	# trainable params	MACs	Inference speed	Mean
VC-1	X	85.8M	16.88G	6.04 ms	80.7
VC-1 + ViT-Adapter	Injector & Extractor	103.2M	26.07G	14.50 ms	80.8
VC-1 + CoIn (Ours)	Injector only	88.9M	19.06G	8.78 ms	84.6

Moreover, through extensive experiments across 12 varied visuo-motor control tasks, we empirically observed that CoIn outperforms ViT-Adapter significantly in (i) computational cost (26.07G $\rightarrow$ 19.06G), (ii) inference speed (14.50 ms $\rightarrow$ 8.78 ms) and in (iii) mean score performance (80.8 $\rightarrow$ 84.6). These empirical results clearly supports our architectural choices behind CoIn and demonstrates the advantages of CoIn over ViT-Adapter for visuo-motor control tasks. We note that computational costs were calculated using ptflops[3] (Sovrasov, 2024) and inference speed was calculated on a single RTX-3090 GPU using a single input image with a resolution of 224 x 224.

In summary, CoIn's contribution is in its optimized and practical architecture tailored towards effectively adapting large-scaled pretrained ViTs for visuo-motor control applications. CoIn differs from ViT-Adapter in terms of both motivation and architectural design, while also demonstrating superior performance in computational cost, inference speed and mean performance.

C. Implementation Details

C.1. Visual encoder

Detailed hyperparameters for finetuning pretrained visual encoders with and without CoIn are listed in Table 8. The same set of hyperparameters is shared across all pretrained visual encoders, regardless of the task and architecture, with the following exceptions: (i) a smaller learning weight is used for CLIP-related experiments, and (ii) different layer-wise learning rate decay values are applied between ViT-based models and ResNet-based models.

C.2. Control policy

We closely follow the architecture and training hyperparameters of the control policy network from prior work (Majumdar et al., 2023; Hansen et al., 2022). Specifically, the control policy network is a 4-layer MLP with 256 hidden units each and ReLU activation. Additionally, the control policy includes a 1D BatchNorm layer at the beginning to normalize the pretrained visual representations. As in VC-1 (Majumdar et al., 2023), we also use frame-stacking, where the visual encoder individually encodes each observation in the stack of recent observations. The control policy then fuses the encoded features using Flare (Shang et al., 2021). Detailed hyperparameters for finetuning the control policy network are listed in Table 9. As with their visual encoder counterpart, the same set of hyperparameters are shared across all tasks regardless of the visual encoder architecture.

C.3. CoIn

Positional embeddings For the multi-scale convolutional feature maps $\mathcal{F}_{\text{conv}} = \{\mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3\}$ within CoIn's Injector module, we assign separate learnable 1D embeddings for each scale which act as positional embeddings. For example, feature map $\mathcal{F}_1$ with scale $H/8 \times W/8$ is combined with a positional embedding $E_{\mathcal{F}_1} \in \mathbb{R}^d$, while feature map $\mathcal{F}_3$ with scale $H/32 \times W/32$ is combined with a different embedding $E_{\mathcal{F}_3} \in \mathbb{R}^d$. These learnable 1D embeddings enable the model to distinguish features extracted from different scales.

[3]Code : https://github.com/sovrasov/flops-counter.pytorch

Table 8. **Visual encoder finetuning hyperparameters.** The same set of hyperparameters is applied across all pretrained ViT and ResNet methods - except for (i) using a smaller learning weight for CLIP-related experiments and (ii) applying different layer-wise learning rate decay values between ViT-based models and ResNet-based models - when performing finetuning regardless of with or without CoIn for all tasks.

Hyperparameter	Value
Optimizer	AdamW
Learning rate	CLIP : 1×10^{-4} others : 1×10^{-3}
Optimizer momentum	$\beta_1, \beta_2 = 0.9, 0.999$
Layer-wise lr decay	ViT : 0.75 ResNet : 1.0
Weight decay	0.05
Batch size	256
Learning rate schedule	cosine decay
Warmup epochs	5
Training epochs	100

Table 9. **Control policy finetuning hyperparameters.** The same hyperparameters are applied across all tasks.

Hyperparameter	Value
Optimizer	Adam
Learning rate	1×10^{-3}
Optimizer momentum	$\beta_1, \beta_2 = 0.9, 0.999$
Hidden units	256, 256, 256
Frames stacked	3
Batch size	256
Training epochs	100

Cross-attention mechanism CoIn's cross-attention mechanism uses deformable attention (Zhu et al., 2020) to address the inherent computational inefficiency of global self-attention where each query token has a global spatial receptive field and examines every key/value token when computing attention weights. This leads to quadratic computation requirements in terms of the total number of query and key tokens. Motivated by the key underlying principle of deformable convolution (Dai et al., 2017), our implementation enables each query patch token (i.e., reference points) in the deformable attention module to selectively focus on a small set of spatially relevant locations (i.e., sampling locations) by predicting a fixed number of sampling offsets respective to the reference point. This selective attention module circumvents the necessity of computing attention weights for every key token for each query token, concentrating instead on a fixed number of key points identified for each query patch token. As a result, this significantly reduces the computational complexity to linear terms relative to the number of query tokens, enhancing the efficiency of the cross-attention process. Therefore, employing deformable attention aligns with our goal in achieving a scalable, fast, and effective cross-attention mechanism for CoIn, while also focusing on spatially relevant local locations.

Architecture configurations For the deformable attention, we fix the number of sampling points to 4, and the number of attention heads to 12 and 16 for ViT-B and ViT-L, respectively. In addition, we downsize the feature embedding size in our Injector module to save computation overhead, where the hidden dimension size is 192 for ViT-B and 256 for ViT-L. We only use a single Injector module in all experiments, as additional Injector modules did not provide additional gains.

D. Main Results

Table 10. **Success rate of each individual task and model.** We present the success rate and standard deviation for each task and model we evaluate during our experiments for Section 5 before aggregating them for each benchmark. All tasks were evaluated with three independent seeds.

Task	Adroit		MetaWorld					DMC				
Model	pen	relocate	hammer	drawer open	button press	bin picking	assembly	walker stand	walker walk	reacher easy	finger spin	cheetah run
VIP Frozen	78.7 ± 2.3	37.3 ± 12.9	94.7 ± 4.6	98.7 ± 2.3	82.7 ± 2.3	93.3 ± 2.3	90.7 ± 6.1	76.9 ± 8.0	47.2 ± 1.7	89.7 ± 4.8	70.2 ± 0.4	38.2 ± 4.3
VIP Finetuned	80.0 ± 0.0	46.7 ± 9.2	98.7 ± 2.3	100.0 ± 0.0	96.0 ± 4.0	88.0 ± 4.0	94.7 ± 9.2	94.5 ± 1.3	87.7 ± 1.6	89.1 ± 2.6	69.8 ± 0.2	70.7 ± 3.4
R3M Frozen	78.7 ± 4.6	44.0 ± 8.0	100.0 ± 0.0	100.0 ± 0.0	74.7 ± 6.1	93.3 ± 2.3	94.7 ± 6.1	88.2 ± 1.0	64.5 ± 6.8	91.0 ± 6.1	68.7 ± 1.0	36.7 ± 3.9
R3M Finetuned	82.7 ± 2.3	74.7 ± 4.6	100.0 ± 0.0	100.0 ± 0.0	84.0 ± 10.6	93.3 ± 2.3	97.3 ± 4.6	96.6 ± 1.3	89.7 ± 1.0	91.8 ± 4.3	68.9 ± 0.8	62.1 ± 1.1
CLIP Frozen	68.0 ± 4.0	9.3 ± 2.3	80.0 ± 4.0	98.7 ± 2.3	56.0 ± 8.0	40.0 ± 6.9	28.0 ± 4.0	43.8 ± 4.6	14.9 ± 1.0	50.5 ± 2.1	63.6 ± 2.4	14.4 ± 1.1
CLIP Finetuned	76.0 ± 4.0	18.7 ± 2.3	90.7 ± 2.3	97.3 ± 2.3	46.7 ± 16.2	65.3 ± 12.9	44.0 ± 6.9	83.8 ± 3.3	49.8 ± 9.0	79.7 ± 4.3	68.9 ± 0.9	31.9 ± 5.4
CLIP Finetuned + CoIn	76.0 ± 4.0	29.3 ± 8.3	98.7 ± 2.3	100.0 ± 0.0	76.0 ± 0.0	90.7 ± 8.3	78.7 ± 4.6	92.6 ± 5.2	77.6 ± 3.5	72.0 ± 3.0	70.6 ± 1.1	42.7 ± 5.7
MVP Frozen	77.3 ± 2.3	38.7 ± 4.6	92.0 ± 6.9	100.0 ± 0.0	85.3 ± 4.6	80.0 ± 4.0	90.7 ± 9.2	82.6 ± 5.7	52.6 ± 7.6	91.7 ± 4.9	70.4 ± 0.3	25.8 ± 7.3
MVP Finetuned	81.3 ± 2.3	82.7 ± 8.3	100.0 ± 0.0	100.0 ± 0.0	89.3 ± 4.6	89.3 ± 9.2	92.0 ± 10.6	96.3 ± 0.9	89.8 ± 1.3	84.4 ± 4.2	69.9 ± 0.8	46.7 ± 2.1
MVP Finetuned + CoIn	78.7 ± 2.3	88.0 ± 6.9	100.0 ± 0.0	100.0 ± 0.0	92.0 ± 8.0	86.7 ± 2.3	96.0 ± 6.9	96.3 ± 0.8	88.9 ± 2.1	98.0 ± 0.5	68.9 ± 3.0	50.7 ± 6.2
VC-1 Frozen	73.3 ± 4.6	26.7 ± 6.1	96.0 ± 4.0	100.0 ± 0.0	81.3 ± 10.1	70.7 ± 4.6	85.3 ± 8.3	75.5 ± 1.7	44.6 ± 3.3	83.1 ± 5.7	70.4 ± 1.3	31.3 ± 4.0
VC-1 Finetuned	74.7 ± 2.3	72.0 ± 8.0	98.7 ± 2.3	100.0 ± 0.0	93.3 ± 4.6	85.3 ± 2.3	92.0 ± 10.6	96.5 ± 0.5	78.4 ± 3.6	83.4 ± 8.2	69.6 ± 0.9	46.6 ± 4.3
VC-1 Finetuned + CoIn	80.0 ± 4.0	74.7 ± 6.1	100.0 ± 0.0	100.0 ± 0.0	93.3 ± 2.3	90.7 ± 2.3	94.7 ± 6.1	95.9 ± 2.1	86.6 ± 3.3	93.6 ± 6.8	69.3 ± 0.6	58.3 ± 8.3

E. Adapter-based Methods Results

Table 11. **Detailed full finetuning performance results against adapter methods.** We report the full performance results of CoIn compared to adapter-based methods for full finetuning, using CLIP and VC-1 with ViT-B across three independent seeds.

Model	Module	# trainable params	Adroit	MetaWorld	DMC	Mean
CLIP	X	85.8M	47.3 ± 3.2	68.8 ± 8.1	62.8 ± 4.6	59.6
	AdaptFormer_{64}	+1.2M	48.0 ± 5.4	71.7 ± 8.0	58.6 ± 3.8	59.4
	RoboAdapter_{64}	+1.2M	47.3 ± 3.2	68.8 ± 5.9	61.4 ± 3.0	59.2
	RoboAdapter_{192}	+3.5M	47.3 ± 1.2	67.5 ± 8.9	62.4 ± 3.7	59.1
	CoIn	+3.1M	**52.7 ± 6.2**	**88.8 ± 3.1**	**71.1 ± 3.7**	**70.9**
VC-1	X	85.8M	73.3 ± 5.2	93.9 ± 4.0	74.9 ± 3.5	80.7
	AdaptFormer_{64}	+1.2M	75.3 ± 3.1	94.1 ± 4.3	77.4 ± 2.7	82.3
	RoboAdapter_{64}	+1.2M	74.7 ± 4.3	94.9 ± 2.9	79.5 ± 2.1	83.0
	RoboAdapter_{192}	+3.5M	75.3 ± 6.2	95.2 ± 4.6	77.1 ± 3.2	82.5
	CoIn	+3.1M	**77.3** ± 5.1	**95.7** ± 2.2	**80.7** ± 4.2	**84.6**

Table 12. **Detailed PEFT performance results against adapter methods.** We report the full performance results of CoIn compared to adapter-based methods for parameter-efficient finetuning, using CLIP and VC-1 with ViT-B across three independent seeds.

Model	Module	# trainable params	Adroit	MetaWorld	DMC	Mean
CLIP	X	–	38.7 ± 3.2	60.5 ± 5.1	37.4 ± 2.3	45.5
	AdaptFormer_{64}	1.2M	41.3 ± 2.3	67.7 ± 8.4	45.0 ± 4.4	51.3
	RoboAdapter_{64}	1.2M	44.7 ± 6.5	61.9 ± 6.5	43.7 ± 3.2	50.1
	RoboAdapter_{192}	3.6M	45.3 ± 5.8	61.9 ± 4.8	43.8 ± 2.6	50.3
	CoIn	3.3M	**51.3**$\pm$ **10.1**	**86.9** $\pm$ **5.5**	**65.4** $\pm$ **4.4**	**67.9**
VC-1	X	–	50.0 ± 5.4	86.7 ± 5.4	61.0 ± 3.2	65.9
	AdaptFormer_{64}	1.2M	68.7 ± 8.4	90.9 ± 5.4	76.8 ± 2.9	78.8
	RoboAdapter_{64}	1.2M	70.0 ± 4.2	93.1 ± 4.9	73.3 ± 3.0	78.8
	RoboAdapter_{192}	3.6M	69.3 ± 5.1	91.5 ± 5.8	75.8 ± 3.9	78.9
	CoIn	3.3M	**66.7** $\pm$ **8.1**	**93.9** $\pm$ **3.1**	**77.8** $\pm$ **3.9**	**79.5**

F. Ablation Results

Table 13. **Detailed performance results for CoIn with larger scale.** We provide the full performance results of CoIn when paired with ViT-B/16 and ViT-L/14 for VC-1 across three independent seeds. We observe that CoIn boosts visuo-motor control task performance for both scales, where ViT-B + CoIn outperforms the larger ViT-L.

Model & Strategy	Backbone Scale & Additional Module	# trainable params	Adroit	MetaWorld	DMC	Mean
VC-1+ Finetuned	ViT-B	85.8M	73.3 ±5.2	93.9 ±4.0	74.9 ±3.5	80.7
	ViT-B + CoIn	88.9M	77.3 ±5.1	95.7 ±2.2	80.7±4.2	84.6
	ViT-L	303.3M	78.7 ±7.6	95.2±4.9	76.3±1.1	83.4
	ViT-L + CoIn	307.7M	76.0 ±6.1	97.6 ±3.1	79.8 ±3.3	84.5

G. Additional Visualization of CoIn

To analyze where our model focuses on the input images, we apply the Attention Rollout technique as described by Abnar & Zuidema (2020); Gildenblat (2020). We first select the attention heads with the maximum attention weights (minimum for DMC) and eliminate 90% of the attention pixels to concentrate on the most significant parts. Figure 6 presents further qualitative examples of these attention rollouts. Overall, the integration of ViT with CoIn demonstrates a relatively precise ability to identify the positions of hands/grippers and objects (Adroit and MetaWorld) as well as the agents (DMC).

Figure 6. **Attention rollout visualization.** Additional qualitative attention map visualization for all tasks.