Voltron-X: Capability-Aware Representation Learning for Robotics

International Journal of Robotics Research
XX(X):1–30

DOI: 10.1177/ToBeAssigned
www.sagepub.com/
SAGE

Siddharth Karamcheti[1], Suraj Nair[2], Annie S. Chen[1], Thomas Kollar[2], Chelsea Finn[1], Dorsa Sadigh[1], and Percy Liang[1]

Abstract

The promise of representation learning is rooted in producing strong initializations – components of broader systems that can be quickly and easily adapted for a spectrum of applications. Realizing this promise is especially important for robotics, as there are a wide range of problems we care about – grasping, visual grounding, and control, amongst others – with high costs for data collection. Yet, existing approaches for visual representation learning over-index on individual problems, tailoring their choice of learning objectives and pretraining data in a way that actively hurts transfer to others. Instead, we argue that learning powerful representations requires both a flexible learning framework and data that capture the capabilities needed for downstream learning. To this end, we introduce **Voltron**, a framework for language-driven representation learning from image and video datasets with associated language annotations. Voltron trades off two objectives to flexibly encode different types of visual features: language-conditioned visual reconstruction to learn low-level visual patterns, and visually grounded language generation to encode high-level semantics. In controlled experiments, we demonstrate that Voltron representations outperform the prior state-of-the-art on a diverse evaluation suite spanning a range of robot learning applications. We then move beyond the controlled setting, and focus on data. Unlike prior work that assume any amount of large, offline data enhances the capabilities of learned representations, we take a more measured approach. We motivate capability-aware data sourcing, an approach for building diverse pretraining mixtures that are particularly useful for transfer *to a wide range of robotics applications*. By explicitly considering the capabilities we want to encode, we construct a clean and compact dataset of fewer than 500K examples – half the size of ImageNet, and 2-3 orders of magnitude less data than used in prior work. We use this data to pretrain and adapt a new suite of models – **Voltron-X** – for two distinct robotics applications: sample-efficient visuomotor policy learning, and open-vocabulary part and object detection. Our results demonstrate the flexibility and efficiency of our approach: Voltron-X outperforms models such as R3M and VC-1 on policy learning, and generalist models such as OWL-ViT on fine-grained detection. We open-source all pretrained models, training code, and our evaluation suite.[1]

Keywords

Visual Representation Learning, Robot Learning, Language for Robotics

1 Introduction

A key challenge in robotics is building perception systems that generalize across a wide range of applications (Weiss et al. 1987; Chaumette and Hutchinson 2006; Levine et al. 2016). Rather than develop such systems from scratch, recent work in robotics propose different approaches for learning visual representations from large offline datasets of diverse human behaviors (Goyal et al. 2017; Grauman et al. 2022), with a goal of learning representations amenable to learning for visuomotor control (Parisi et al. 2022; Nair et al. 2022; Radosavovic et al. 2022; Ma et al. 2022). Yet, robot learning is a discipline spanning a *diverse spectrum of problems*, each with their own requirements on the types of visual features or priors captured in learned representations. For example, a perception system for predicting grasp proposals on novel objects (Saxena et al. 2008; Mahler et al. 2017) must capture low-level features of shapes and objects such as texture, color, and material, while a system for instruction following (Tellex et al. 2011) or intent inference (Hauser 2012; Javdani et al. 2018) may need higher-level features that capture how a scene changes over time. Because of these distinct needs, tailoring learning objectives *and* pretraining data to individual applications can come at a steep cost: that of transfer to other tasks and applications. Instead, we argue that learning generalizable representations requires two components: a framework for representation learning that enables encoding different visual priors, as well as a method for sourcing data that reflect the capabilities necessary for a diverse set of applications. Developing a pipeline for *capability-aware representation learning* (Fig. 1) is the core question motivating this work.

As a first step, we show the harms of building representations around a single application. We evaluate two recent approaches for learning visual representations against a broad suite of applications beyond control, each reflecting different inductive biases in what the learned representations

[1]Department of Computer Science, Stanford University
[2]Toyota Research Institute

Corresponding author:
Siddharth Karamcheti, Stanford University
Email: skaramcheti@cs.stanford.edu

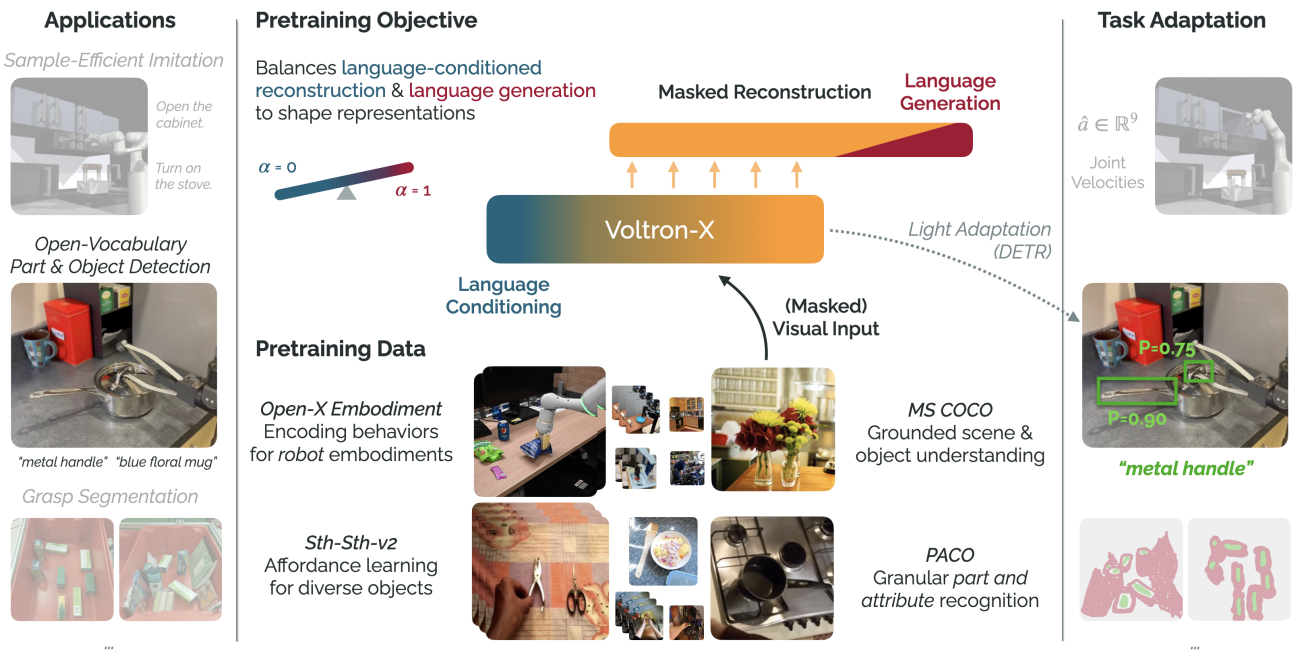

Figure 1. Voltron-X – Capability-Aware Representation Learning. We introduce an approach for sourcing compact datasets for visual representation learning that enable transfer to a broad range of robotics applications (**Left**). We do this by identifying a set of "capabilities" – properties or priors that a representation should be able to learn from a given dataset – which inform data curation. We train visual representations on the resulting mixture using **Voltron** (§3), our framework for language-driven representation learning (**Middle**). Our learned representations are flexible, and can be quickly and easily adapted for a spectrum of downstream tasks (**Right**).

capture. Masked Visual Pretraining (MVP; Radosavovic et al. 2022) proposes using masked autoencoding (He et al. 2022) to prioritize visual reconstruction from heavily masked video frames, encoding representations that facilitate per-pixel reconstruction. Separately, Reusable Representations for Robotic Manipulation (R3M; Nair et al. 2022) eschews pixel reconstruction for two contrastive learning objectives: time contrastive learning (Sermanet et al. 2018) and video-language alignment. We show that MVP performs well on problems such as grasp prediction, but struggles at encoding higher-level features for problems such as language-conditioned imitation learning. Conversely, R3M excels at encoding high-level behavior, but degrades completely in settings requiring fine-grained reasoning.

Motivated by this discrepancy, we introduce **Voltron**, a framework for language-driven representation learning that is amenable to capturing different types of visual priors. Our key insight is that language supervision is a powerful signal for modulating the abstractions captured in learned representations. Voltron models take images or videos and associated language annotations as input to a masked autoencoding pipeline, reconstructing one (or more) frames from a masked context. Depending on a tunable probability α, we either condition on $(\alpha = 0)$, or generate $(\alpha > 0)$ the associated caption. Explicitly *conditioning* on words in different contexts allows for low-level pattern recognition at the local, spatial level, while *generating* language from our learned visual encoding allow us to infer higher-level features around affordances and intents.

To evaluate Voltron and other visual representation learning approaches, we assemble a new evaluation suite (depicted in Fig. 2) spanning five problem domains within robotics: 1) dense segmentation for grasp affordance prediction (Zeng et al. 2017), 2) object detection from referring expressions (e.g., "the blue coffee mug to the left of the plate") in cluttered scenes (Wang et al. 2021), 3) imitation learning for visuomotor control (in simulation) (Nair et al. 2022), 4) learning multi-task language-conditioned policies for real-world manipulation (Stepputtis et al. 2020) (on a real-world Franka Emika fixed-arm manipulator), and 5) zero-shot intent scoring (Javdani et al. 2018; Chen et al. 2021a). We choose these tasks for their broad coverage; tasks such as grasp affordance prediction and referring expression grounding require reasoning over low-level spatial features, while language-conditioned imitation and intent scoring require a deeper understanding of semantics. Through experiments controlling for pretraining data and model capacity, we show that Voltron representations strictly outperform both MVP and R3M across *all* evaluation domains.

While our results and controlled experiments establish Voltron as a general framework for learning flexible visual representations capable of encoding different types of visual priors, they fail to explore the second key component in developing powerful representations – data. We motivate *capability-aware data sourcing*, an approach for building diverse pretraining mixtures that enable transfer to a wide range of robotics applications. We first demonstrate the need for such an approach through a negative result; we show that naively scaling Voltron on large "in-the-wild" datasets used in prior work (e.g., Ego4D; Grauman et al. 2022) leads to significantly degraded representations, even as we train on more samples. Our analysis reveals that dataset artifacts such as mislabeled captions and motion blur actively impede learning – not just for Voltron representations, but for other approaches as well (Dasari et al. 2023). Instead, we frame data sourcing through the lens of individual capabilities we want to encode – e.g., generalization across different embodiments, fine-grained recognition of materials and attributes, etc. – and

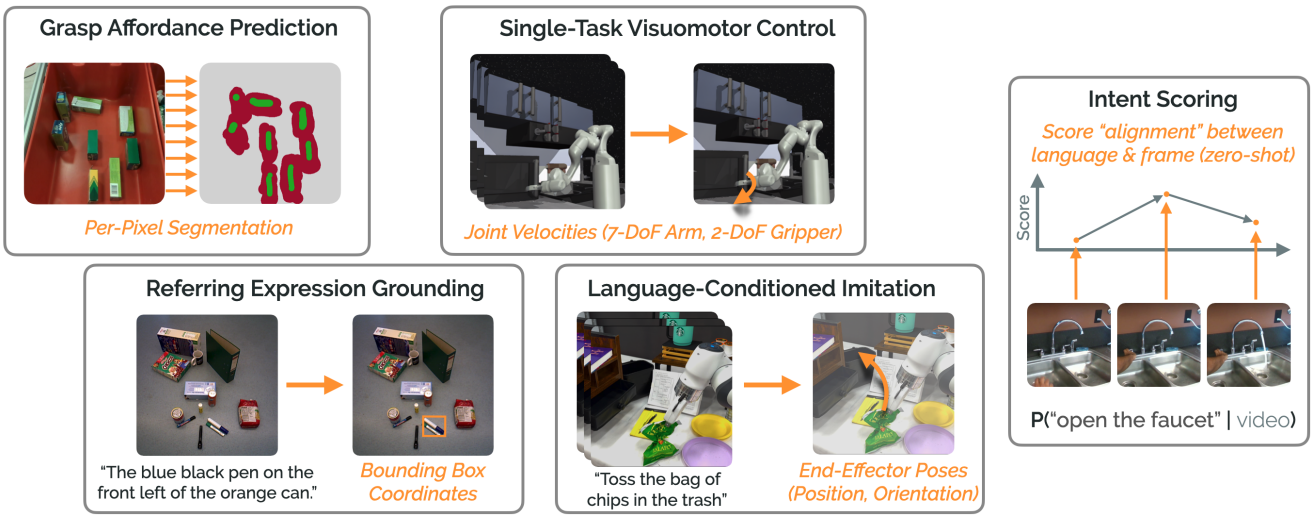

Figure 2. Voltron Evaluation Suite. We introduce a suite of evaluation problems spanning *five applications within robotics*, including grasp affordance prediction, referring expression grounding, single-task visuomotor control (in simulation), language-conditioned imitation learning (on a real robot), and intent scoring.

identify a clean and compact mixture of data of fewer than 500K examples sourced from existing datasets that enable a diverse range of applications.

We use this dataset to pretrain and adapt a new suite of models – **Voltron-X** – in a case study that considers distinct robotics applications: sample efficient visuomotor policy learning, and open-vocabulary part and object detection. We choose these specific applications because they highlight the range of visual priors and capabilities a pretrained representations must encode. Visuomotor policy learning requires representations that capture high-level semantic features around behaviors and affordances, while fine-grained open-vocabulary detection requires grounded visual representations that are granular, picking up on features at the level of individual parts and material composition of objects. Our results demonstrate both the versatility of Voltron as a framework and capability-aware data sourcing as a way to craft diverse and compact pretraining mixtures that enable transfer to a spectrum of robotics applications. Voltron-X outperforms larger models trained on more data and compute; On sample-efficient policy learning we outperform both R3M and VC-1 (Majumdar et al. 2023), while on fine-grained detection, we outperform both closed set detectors trained specifically for this task, as well as generalist open-vocabulary models such as OWL-ViT (Minderer et al. 2022).

2 Related Work

An established body of work in robot learning studies learning visual state representations for control. Prior approaches learn representations from *in-domain* data taken directly from the target environment (and corresponding task); these techniques range from using data augmentation (Laskin et al. 2020; Srinivas et al. 2020; Kostrikov et al. 2021; Pari et al. 2022) to modeling forward dynamics (Gelada et al. 2019; Hafner et al. 2020) to using task-specific information (Jonschkowski and Brock 2015; Zhang et al. 2021). Unlike these approaches, we move beyond task-specific data, instead leveraging large, accessible datasets such as videos of humans performing everyday tasks, or recently released collections of robot demonstrations paired with language instructions that span multiple tasks and embodiments (Padalkar et al. 2023). Work in this paradigm has exploded in recent years. A number of approaches find that existing representations such as features from models trained on ImageNet (Deng et al. 2009), or features from CLIP (Radford et al. 2021) enable more efficient learning (Shah and Kumar 2021; Khandelwal et al. 2021). More recently, multiple approaches have shown increased dividends in applying such representations to visuomotor control, for example by combining features at different layers of pretrained ResNets (Parisi et al. 2022) or by pretraining such representations on human videos, conjecturing that such data captures features useful for robotic manipulation (Nair et al. 2022; Xiao et al. 2022; Radosavovic et al. 2022; Ma et al. 2022; Majumdar et al. 2023). Missing from these approaches however is a notion of semantics; works such as MVP (Xiao et al. 2022; Radosavovic et al. 2022) purely learn to perform masked reconstruction, and even works that leverage *some* temporal and linguistic signals do so in a limited way (Nair et al. 2022; Ma et al. 2022). Instead, our work is motivated by the hypothesis that language understanding – via conditioning *and* generation – is an essential component for learning generalizable representations. It is not enough that a representation summarizes an observation; instead, for generalization to new contexts, it must capture how observations (and *changes* thereof) relate to higher-level semantic abstractions.

Voltron aims to do this with its language-driven representation learning objective: by jointly modeling image frames *and* language, we enable a range of capabilities, from producing representations of single images in isolation, to providing the capability to *generate* language grounded in visual contexts. We demonstrate the benefits of language-driven learning in our evaluation (see §5): in head-to-head comparisons controlling for data and model capacity, Voltron models strictly outperform prior approaches across *all* evaluation domains. We further demonstrate the flexibility of Voltron representations in §8; with the right pretraining data and light adaptation, we can transform Voltron representations

into fully-fledged systems for individual tasks such as open-vocabulary part-and-object detection, outperforming existing state-of-the-art models with far less data and compute.

Learning Multimodal Foundation Models. Our work draws further inspiration from a wave of progress in multimodal foundation models (Bommasani et al. 2021) such as CLIP, Multimodal Masked Autoencoders (M3AE), Flamingo, CoCa, and Gato, amongst many others (Radford et al. 2021; Geng et al. 2022; Alayrac et al. 2022; Yu et al. 2022; Reed et al. 2022; Lu et al. 2023; Aghajanyan et al. 2022). These approaches highlight the myriad benefits of multimodal pretraining: language supervision works to enrich visual representations (even *in the absence of language downstream*), while visual supervision similarly enriches language representations (Lu et al. 2019; Singh et al. 2022). Of the many capabilities afforded by these models, many have applications in embodied AI and robotics. CLIP representations have shown to be effective in applications to various robotics tasks (Shridhar et al. 2021; Khandelwal et al. 2021; Cui et al. 2022), while multimodal transformer models have proven effective initializations for training control policies (Reid et al. 2022; Liu et al. 2022). These approaches are similar to Voltron in their use of visual and language; where Voltron differs, however, is in our novel representation learning objective that balances language conditioning and generation, enabling learning representations that transfer to a wide range of applications within robotics.

3 Voltron – Language-Driven Learning

We assume access to a dataset of images or videos (frames) paired with natural language annotations; in each pair of visual context and language (v, c), language can take the form of a caption (e.g., "peels the carrot" in Fig. 3), narration, or even coarse textual label such as a category or attribute. We assume each visual context $v \in \mathbb{R}^{T \times H \times W \times C}$ consists of one or more frames $v = [o_1, \ldots, o_T]$, where each frame $o_i \in \mathbb{R}^{H \times W \times C}$ is RGB-encoded. We tokenize and one-hot encode each utterance into a vocabulary V of cardinality $|V|$, padding to a max length L such that $c \in \mathbb{R}^{L \times |V|}$. We define a <NULL> token (separate from the <PAD> token) as a placeholder for an empty language context. Furthermore, following the MAE work, we define a visual masking function $\texttt{Mask}(v, \gamma) \rightarrow (v_{\text{visible}} \in \mathbb{R}^{(1-\gamma)(T \times H \times W \times C)}, v_{\text{masked}} \in \mathbb{R}^{\gamma(T \times H \times W \times C)})$ that partitions the regions of a video into a set of visible and masked-out regions subject to a fixed masking ratio γ. This mask is held constant across timesteps in a given clip. We sample a mask once, and apply it uniformly across *all* frames in the video to prevent leakage (Tong et al. 2022); if the masks were sampled independently, a masked region in one frame could be visible in another, allowing the encoder to "cheat" by looking ahead.

3.1 Voltron – Core Components

A Voltron model comprises 1) a *multimodal encoder* that takes in a visual context and (optional) language utterance producing a dense representation, 2) a *visual reconstructor* that attempts to reconstruct the masked-out visual context from the encoder's representation of what is visible, and 3) a *language generator* that attempts to generate the language annotation given the encoded context. The visual reconstructor and language generator crucially act to shape the representations by first erasing portions of a (v, c) pair, then attempting to reconstruct the missing parts; we show in our experiments (see §5) that this bottleneck helps focus on more low-level features when we favor reconstruction, and more higher-level semantic features when we favor language generation. We step through each component below.

Multimodal Encoder: $\text{E}_\theta(\tilde{v}, u) \rightarrow h \in \mathbb{R}^{S \times d}$

The multimodal encoder (Fig. 3; lower half in blue and orange) is the core of a Voltron model. It takes as input $(\tilde{v}, u)$ where $\tilde{v} \in \{v_{\text{visible}}, v\}$ denotes either the *masked* or *unmasked* (full) visual context respectively, and u represents a (possibly <NULL>) utterance to condition on. As output, the encoder produces a dense representation $h \in \mathbb{R}^{S \times d}$ where S denotes the number of encoded regions, and d is a hyperparameter denoting the dimensionality of the representation. Keeping with the original MAE work, we divide each image $o_i \in \mathbb{R}^{H \times W \times C}$ into a set of non-overlapping regions R, where each region is a $p \times p$ patch; this results in $|R| = HW/p^2$ regions. Given a k-frame context, $S = (1 - \gamma)k|R|$.

Visual Reconstructor: $\text{R}_\theta(h) \rightarrow \hat{v}_{\text{masked}} \in \mathbb{R}^{\gamma(k \times H \times W \times C)}$

The visual reconstructor (Fig. 3; upper half in orange) takes as input the encoded representation of the *visible* visual context $h = \text{E}_\theta(v_{\text{visible}}, c)$. It attempts to reconstruct the missing visual regions v_{masked}, conditioned on language context c, producing a prediction $\hat{v}_{\text{masked}}$. Following prior work, the elements of $\hat{v}_{\text{masked}}$ are the normalized pixel targets from the original image. We use mean-squared error as the reconstruction loss $\mathcal{L}_{\text{reconstruct}}(\theta)$.

Language Generator: $\text{G}_\theta(h) \rightarrow \hat{c} \in \mathbb{R}^{L \times C}$

The language generator (Fig. 3; upper half in red) takes the encoded representation of the *visible* context and the <NULL> language token, $h = \text{E}_\theta(v_{\text{visible}}, \texttt{<NULL>})$. It generates the language annotation, producing $\hat{c} \in \mathbb{R}^{L \times |V|}$, with each of the L elements corresponding to a probability distribution over the vocabulary. We use the negative log-likelihood of the annotation c under the generator as our loss $\mathcal{L}_{\text{generate}}$.

The language generator crucially takes the <NULL> token as input instead of the annotation c; inputting the same c that the generator is trying to output can lead to trivial collapse where the encoder learns to memorize the tokens to aid the generator. As a result, for each example during training we need to *either* condition *or* generate language; this further motivates the parameter α in Fig. 3 and in the training objective.

3.2 Balancing Reconstruction & Generation

The Voltron learning objective trades off language-conditioned *reconstruction* and visually-grounded *language generation* to shape the features captured by the encoder's learned representation. The reconstruction objective prioritizes low-level spatial information conducive to filling in missing textures, colors, or edges; conversely, the generation objective captures higher-level semantics, encouraging the encoder to encode features that are predictive of the language caption. We make this tradeoff explicit by minimizing the

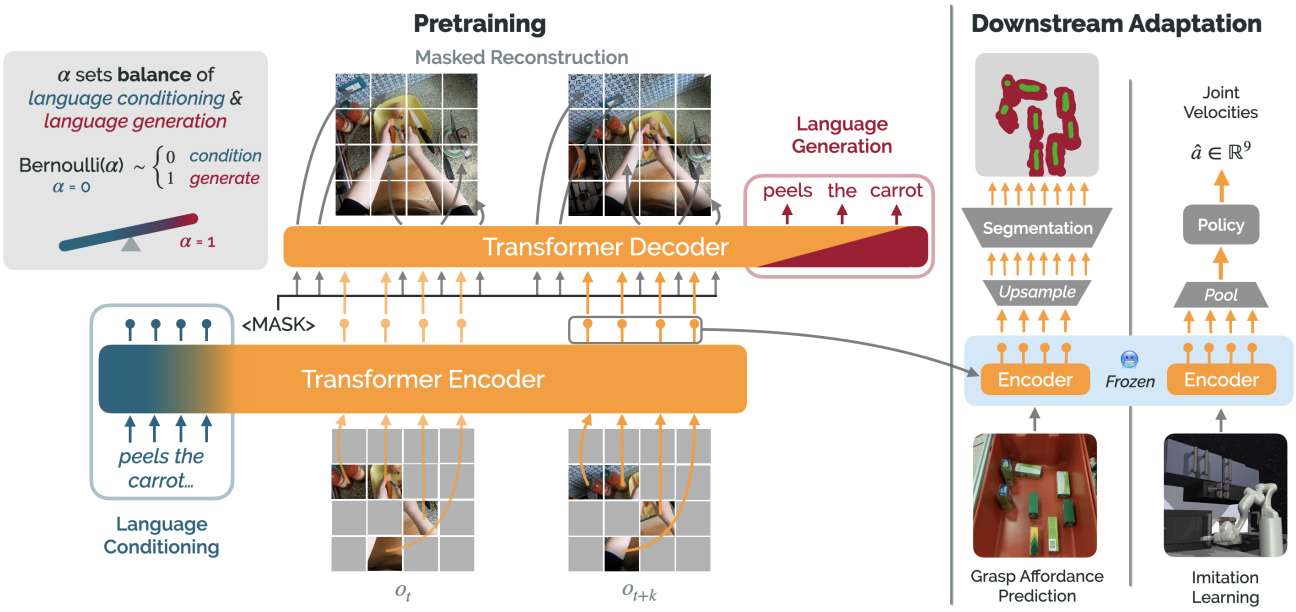

Figure 3. The Voltron Framework. Central to our approach is *language-driven learning* on top of a masked autoencoding backbone. We incorporate language in two ways, following §3.2: 1) as a *conditioning variable* fed to a multimodal encoder that also encodes one or more video frames, or 2) as a *generation target* for the language generator **[Left]**. During downstream evaluation, we use the (frozen) outputs from the encoder, adapting evaluation-specific "heads" on top **[Right]**.

following loss, parameterized by $\alpha \in [0, 1]$:

$$\mathcal{L}(\theta) = \mathcal{L}_{\text{reconstruct}}(\theta) + \mathcal{L}_{\text{generate}}(\theta)$$
$$= \begin{cases} \text{MSE}(v_{\text{masked}}, \text{R}_\theta(\text{E}_\theta(v_{\text{visible}}, c))) & \text{if } z = 0 \\ \text{MSE}(v_{\text{masked}}, \text{R}_\theta(\text{E}_\theta(v_{\text{visible}}, \texttt{<NULL>}))) & \text{if } z = 1 \\ \quad + \text{NLL}(c, \text{G}_\theta(\text{E}_\theta(v_{\text{visible}}, \texttt{<NULL>}))) & \end{cases}$$
$$\text{and } z \sim \text{Bernoulli}(\alpha)$$

For each example (v, c) seen at training, we draw $z \sim \text{Bernoulli}(\alpha)$: with $z = 0$ we *condition* on the original language utterance, while with $z = 1$, we *generate* the original language utterance, conditioning the encoder on the <NULL> token. We limit our exploration to at most two frame contexts $k = 2$ due to computational cost; even four frame contexts exceed the memory on the compute available to us. In selecting the two frame contexts, we sample at least five frames from each video clip in our dataset (with random intervals between). We enforce a heuristic such that the first frame in each context comes from the first 20% of the clip, with the other frame appearing in the remaining 80%.

Driven by the hypothesis that different values of α and frame-contexts k shape the balance of low-level and high-level features in our representations, we evaluate three different instantiations of the Voltron framework:

- $\mathcal{V}$**-Cond**: $\alpha = 0$, $k = 1$ *single-frame conditioning*.
- $\mathcal{V}$**-Dual**: $\alpha = 0$, $k = 2$ *dual-frame conditioning*; identical to $\mathcal{V}$**-Cond** but trained on dual-frame pairs (initial frame, random subsequent frame).
- $\mathcal{V}$**-Gen**: $\alpha = 0.5$, $k = 2$; condition *and* generate with equal probability, trained on dual-frame pairs.

4 Implementation & Reproducibility

In addition to our framework, a core contribution of this work is a comprehensive set of controlled experiments. To do this, we *reimplement* both MVP and R3M using code released by the authors, controlling for the pretraining data (at the level of the individual frames seen per epoch) and model capacity.

Baselines – Preliminaries. Throughout this work, we have mentioned both MVP and R3M in terms of their tradeoffs; here, we make their pretraining objectives explicit. Both prior approaches use video datasets, but only learn *single-frame encoders*, choosing to use the video structure in different ways (detailed below). Of the two approaches, we note that only R3M uses language supervision.

MVP follows a masked autoencoding backbone, similar to that depicted in Fig. 3 (without language). MVP does not offer any special consideration to the temporal structure of videos, instead treating each frame in the dataset as as standalone input. Given a single frame, MVP masks out regions subject to a fixed mask ratio γ (same as in Voltron), encoding the visible context with a Transformer encoder, then attempting to reconstruct the missing context with a separate Transformer decoder – also using mean-squared error for reconstruction.

R3M is different in that it does not contain a reconstruction component, instead combining *two contrastive objectives* on top of a single-frame visual encoder – time contrastive learning (Sermanet et al. 2018) and image-language temporal alignment (Radford et al. 2021; Nair et al. 2021). These objectives explicitly use the *temporal* structure of videos. Given an encoding of a visual context, the time-contrastive objective seeks to maximize the score of encodings between frames close together in time (e.g., within a few frames of each other), contrasted against frames from the same video that are further away. R3M also *uses language supervision*. Given a separate encoder that fuses a language caption with the encoding dual-frames contexts (consisting of an initial and subsequent frame) the image-language alignment objective attempts to assign scores that capture "task progress:" the score of a subsequent frame occurring later in a video subject to a language caption should be higher than the score of a frame occurring earlier. The two key differences between

Voltron and R3M are 1) using visual reconstruction as a dense objective vs. time contrastive learning, and 2) explicitly conditioning on or generating language in Voltron.

Pretraining Dataset Construction. We use Something-Something-v2 (Sth-Sth; Goyal et al. 2017) as our controlled pretraining dataset, motivated by prior work (Shao et al. 2020; Chen et al. 2021a; Xiao et al. 2022). All models see the *exact same image frames*. We extract 5 frames per video, per training epoch to ensure we are learning from multiple visual inputs of the same context and to facilitate R3M's time contrastive learning objective (Sermanet et al. 2018); we serialize the processed frames, and store index files with the frame indices per epoch.

Data-Equivalent Reproductions. Though prior works release trained model artifacts, they do not provide sufficient details for reproduction, such as the exact frames sampled from videos, preprocessing applied, or hardware/compute used. We thus reimplement MVP and R3M in a controlled setting on Sth-Sth using the released code from the original papers where possible and clarifying additional details with the authors directly as needed. We implement all models with a Vision Transformer (ViT) backbone and additionally implement R3M with a ResNet-50 backbone based on discussions with the authors of the original work who suggested investigating the differences in inductive bias between ResNets and ViTs (Raghu et al. 2021). We use the ViT-Small architecture (Wightman 2019), with patch size $p \times p = 16 \times 16$. We refer to our reproductions as "R-MVP," "R-R3M (ViT-S)," and "R-R3M (RN-50)."

We pretrain all models in this section on TPU v3-8 compute, generously granted to us by the TPU Research Cloud program (TRC). We run 400 epochs of training for all models with a batch size of 1024, each epoch comprised of a pass through 844K frames (168K clips in Sth-Sth, 5 frames per clip). We do not use dropout or data augmentation. All modeling and reproducibility details are in our open-source training code.

Additional Comparisons. We further contextualize our results by evaluating the official R3M and MVP models released in the original works. We note that the released R3M model uses an unspecified subset of the Ego4D dataset (Grauman et al. 2022), comprised of over 3000 hours of videos, spanning over 3M individual clips (constituing a dataset **more than 20x** larger than that used in this work). The released MVP also uses an unspecified subset of Ego4D, but add Sth-Sth, Epic-Kitchens, and more (Damen et al. 2018; Shan et al. 2020), while also scaling models up to 86M and 307M parameters, (**4-10x** the size of ViT-Small). We also evaluate OpenAI's CLIP model (ViT-Base) as a strong baseline that leverages language supervision. We refer to these models as *"R3M (Ego4D)," "MVP (EgoSoup),"* and *"CLIP (ViT-B),"* following naming conventions from the original work and denote them with *gray text* and dashed lines.

Voltron Architecture Details. Voltron follows the masked autoencoding pipeline detailed above, with simple extensions for incorporating language. We implement the Voltron encoder E_θ by jointly embedding the language u and visual inputs v_{visible} with a Transformer (Vaswani et al. 2017). We initialize language embeddings from DistilBERT (Sanh et al. 2019), learning a separate linear projection into the encoder's embedding space, similar to R3M. For the visual reconstructor R_θ and language generator G_θ, we use a separate Transformer with a small addition to enable language generation. In a standard MAE decoder, patches are generated independently, attending to all patch embeddings from the encoder. To enable generation, we append a causal (lower triangular) attention mask for preventing our language decoder from "peeking" at the future inputs to generate (visualized by the red triangle in Fig. 3). This is akin to prefix language modeling (Raffel et al. 2019); all embeddings can attend to the visual inputs (as in a traditional MAE decoder), but language embeddings can only attend to the preceding language input.

Voltron uses a combination of different language objectives on top of the standard MAE pipeline, adding complexity. To help ensure stable and reliable training, we follow best practices from the NLP community and make a series of small changes to the Transformer architecture including: 1) switching the default LayerNorm to root-mean square normalization (Zhang and Sennrich 2019; Narang et al. 2021) (stability, no learned parameters), 2) switching from the default GELU to the more performant SwishGLU activation (Shazeer 2020; Chowdhery et al. 2022) (performance), and 3) adopting LayerScale for scaling down the magnitude of each residual connection (Touvron et al. 2021; Karamcheti et al. 2021a) (prevents overflow). We find that these changes do not change downstream evaluation results, but significantly improve training stability. We present further detail in §A.1.

Adapting Representations. Unfortunately, there is not a standard way to extract representations from learned Vision Transformer encoders, especially for those trained via masked autoencoding. However, Zhai et al. (2022) suggest that multiheaded attention pooling (MAP; Lee et al. 2018) is a strong and versatile approach. We choose to use MAP as the sole feature extraction approach in all our ViT experiments, finding it to *universally improve performance for all ViT models*, relative to the "default" extraction approaches suggested in prior work. Notably, we find that just switching to MAP-based extraction over the procedure used in the original MVP work *almost doubles success rate* on visuomotor control tasks; we provide full experiment details and analysis in §C.2. We also use MAP when evaluating *CLIP (ViT-Base/16)* and *MVP (EgoSoup)* for the strongest possible comparison.

5 Evaluation Suite: Construction & Results

We outline our evaluation suite (Table 1) comprised of five problem domains within robotics. Each evaluation consists of *adaptation data* and *evaluation metrics*. The adaptation data consists of visual input(s) (as RGB) and in some cases, language (e.g., an instruction for language-conditioned imitation). We evaluate representations from Voltron and various baseline models by *freezing the pretrained vision and language encoders*, instead *adapting* evaluation-specific "heads"(lightweight networks) on top of the extracted representations. We choose evaluations that capture different types of visual understanding; in the following sections, we motivate each application and provide experimental results.

5.1 Grasp Affordance Prediction

We consider the problem of grasp affordance prediction: given an image of a set of objects on a cluttered workspace, predict

Table 1. Summary of Evaluation Suite & Results. While some of our evaluation domains use language input, grasp affordance prediction and single-task visuomotor control *do not*. Voltron models obtain strong performance over *all applications*, whereas R-R3M and R-MVP exhibit variable performance depending on the application subset.

	Input	Dataset Size	Best Model
Grasp §5.1	Single Frame	1470	$\mathcal{V}$**-Cond**
Referring Expressions §5.2	Single Frame, **Language Expression**	260K	$\mathcal{V}$**-Cond**
Single-Task Control §5.3	Frame History	$n \in [5, 10, 25]$ Demos	$\mathcal{V}$**-Dual**
Language-Conditioned Imitation §5.4	Frame History, **Instruction**	100 = 5 x 20 Demos	$\mathcal{V}$**-Dual**
Intent Scoring §5.5	Frame History, **Language Intent**	N/A (Zero-Shot)	$\mathcal{V}$**-Gen**

a dense segmentation mask corresponding to "graspable" and "non-graspable" locations for a suction-based gripper.

Figure 4. Grasp Affordance Prediction (ARC Grasping; Zeng et al. 2017). Given objects in cluttered bins, segment the image corresponding to "graspable" (green), vs. "non-graspable" (red) regions; note that regions are labeled for a *suction gripper*.

Motivation. Grasp affordance prediction from visual input is a foundational task in robot learning, and is a key component of many modular systems (Bohg et al. 2013; Correll et al. 2016). Including this evaluation allows us to probe the low-level spatial features retained by various representations.

Evaluation Details. We specifically consider the problem as formulated in the Amazon Robotics Challenge Grasping Dataset (ARC-Grasping) introduced by Zeng et al. (2017). We choose this dataset over alternatives as it is readily available and consists of 1800+ images of multiple real-world objects in cluttered bins (Fig. 4; left). We focus on the RGB-only, suction-grasping split of the dataset. We implement models for grasp affordance prediction following recent work on semantic segmentation with Transformers (Zheng et al. 2021; Strudel et al. 2021; Bao et al. 2022), specifically by introducing a Progressive Upsampling (SETR-PUP) head on top of our frozen visual features. We omit results from all ResNet models – R-R3M (RN-50) and *R3M (Ego4D)*; unfortunately, training with simple PUP-style on the final ResNet-50 7×7 spatial grid did not converge, possibly indicating a need for more complex architectures with significant added parameters (beyond the scope of this work). As this task only takes a single frame as input, we do not evaluate $\mathcal{V}$**-Dual** and $\mathcal{V}$**-Gen**. Following the original work, we report average precision at various confidences: Top-1 precision, Top-1% precision, and Top-5% precision. We select models via 5-fold cross validation. This task *does not have a language component*. We provide additional details around the adaptation procedure in §A.4.

Experimental Results. Looking at Table 2, representations from MVP and Voltron models perform well across the board, while contrastive representations (e.g., from CLIP and R-R3M) perform quite poorly. Interestingly, $\mathcal{V}$**-Cond** outperforms R-MVP and *MVP (EgoSoup)* on this task, *despite the absence of language input*, demonstrating that language supervision improves low-level feature learning, even compared to larger models trained on more data.

5.2 Referring Expression Grounding

Given a cluttered scene and language expression, the goal is to predict a bounding box around an object (e.g., "the blue black pen on the front left of the orange can" in Fig. 5; middle).

Motivation. Capturing object-centric priors and spatial relationships is a desirable skill for any robot system. Furthermore, this is a *language-conditioned* task, allowing us to evaluate the impact of pretraining with language.

Evaluation Details. We use the OCID-Ref Dataset (Wang et al. 2021) grounded in scenes that are representative of robotics settings; other datasets such as RefCoCo (Yu et al. 2016) are grounded in more global scenes (e.g., multiple humans playing frisbee on a field) that are less informative for robot learning. OCID-Ref also provides splits based on the clutter level of the underlying scene, letting us further evaluate robustness. We regress bounding box coordinates directly from our frozen features using a shallow MLP. All approaches condition on language (see expressions in Fig. 5), using the given language encoder where possible. This means using the multimodal encoder for $\mathcal{V}$**-Cond** and the default learned text encoder for CLIP or R3M. However, for approaches that only learn visual representations (e.g., MVP), we append pretrained language features from DistilBERT – the same language model used to initialize Voltron. We note again that we omit ResNet results; though this task did not require upsampling, we find trained models obtained no better than random performance, again indicating a need for a more sophisticated adaptation architecture (beyond the scope of

Table 2. Results on Grasp Affordance Prediction. We report average precision at various confidence intervals following the original procedure described in Zeng et al. (2017).

	Arch.	Top 1	Top 1%	Top 5%
R-R3M	ViT-S	40.38	40.55	28.66
R-MVP	ViT-S	72.94	61.47	39.77
$\mathcal{V}$**-Cond** [Ours]	ViT-S	**85.15**	**80.71**	47.45
$\mathcal{V}$**-Cond** [Ours]	ViT-B	**90.00**	**82.44**	**62.33**
CLIP	*ViT-B*	*43.20*	*44.11*	*29.66*
MVP (EgoSoup)	*ViT-B*	*77.49*	*72.87*	*51.28*

Figure 5. Referring Expression Grounding (Object Detection) from the OCID-Ref Dataset (Wang et al. 2021). Given a referring expression in natural language, the goal is to predict the bounding box coordinates around the respective object. An important feature of OCID-Ref are the various dataset splits, corresponding to three increasing amounts of clutter, depicted left-to-right.

this work). We report average precision at 0.25 IoU for each split following the evaluation procedure outlined in Wang et al. (2021). Additional details are in §A.4.

Experimental Results. Results for each model across the various clutter splits are in Table 3. Voltron models are especially strong, vastly outperforming R-MVP by 40% and R-R3M by over 25% on all splits, showing that multimodal pretraining – even just conditioning on language when optimizing for masked reconstruction – can lead to substantial gains on downstream multimodal tasks. We isolate the performance gains of Voltron models to the multimodal encoder that learns *fused* embeddings of vision and language, allowing language to shape the visual representations during pretraining. In contrast, R3M, and CLIP models learn *independent* text encodings that are only fused during adaptation. This is even worse for MVP: these models need to learn to fuse their strong visual embeddings with the language embeddings from a completely different model (DistilBERT).

5.3 Single-Task Visuomotor Control

Given a dataset of demonstrations, learn a policy for a given task, predicting continuous joint actions from visual observations of the scene from an external camera and proprioceptive state.

Motivation. Imitation learning for visuomotor control has been the de-facto evaluation for prior work (Parisi et al. 2022; Nair et al. 2022; Radosavovic et al. 2022), giving us the closest comparison to the evaluations used in MVP and R3M. This evaluation focuses on *sample-efficient generalization*, measuring how well visual representations help in learning policies from limited demonstrations $n \in \{5, 10, 25\}$.

Evaluation Details. We look at policy learning in the Franka Kitchen simulation environments as defined by Nair et al. (2022). This domain consists of 5 tasks, with 2 distinct camera viewpoints (Fig. 6). We learn shallow MLP policy heads via behavioral cloning that predict 9-DoF joint velocities (7 joints, 2 gripper) from our (frozen) visual features and proprioceptive state. We follow the R3M evaluation, reporting average success rates for each setting with n demonstrations across the 5 tasks, 2 viewpoints, and 3 random seeds. We train separate policies per task, with *no language conditioning* – using the exact code provided by Nair et al. (2022).

Experimental Results. Most approaches perform similarly across the various number of training demonstrations (Fig. 6; right). However, we see some promising trends; Voltron models perform better than both baselines, with approaches that learn from multiple frame contexts **$\mathcal{V}$-Dual** and **$\mathcal{V}$-Gen** showing *significant* improvements over single-frame approaches. Yet absolute success rates are low; while good visual representations can help, learning closed-loop policies from limited data remains an open challenge.

5.4 Real-World Language-Conditioned Imitation Learning

Given a dataset of language instructions (e.g. "throw the bag of chips away") paired with demonstrations (in a real-world tabletop setting), learn an instruction following policy via behavioral cloning. Fig. 7 depicts the real-world environment.

Table 3. Results on Referring Expression Grounding. We report average precision @ 0.25 IoU following Wang et al. (2021) (OCID-Ref). This is a *language-conditioned* task; across various clutter levels, Voltron models are substantially more performant than baselines, as well as models trained on more data and with alternative language supervision (e.g., CLIP).

	Architecture	Total	Min Clutter	Med Clutter	Max Clutter
R-R3M	ViT-Small	63.30	63.87	68.34	55.33
R-MVP + DistilBERT	ViT-Small	49.58	50.98	53.83	41.94
$\mathcal{V}$-Cond [Ours]	ViT-Small	**89.38**	**85.88**	**95.39**	**89.12**
$\mathcal{V}$-Cond [Ours]	ViT-Base	**90.77**	**87.56**	**96.58**	**90.17**
CLIP	*ViT-Base*	*68.35*	*67.01*	*76.61*	*60.33*
MVP (EgoSoup) + DistilBERT	*ViT-Base*	*49.25*	*51.46*	*52.15*	*40.50*

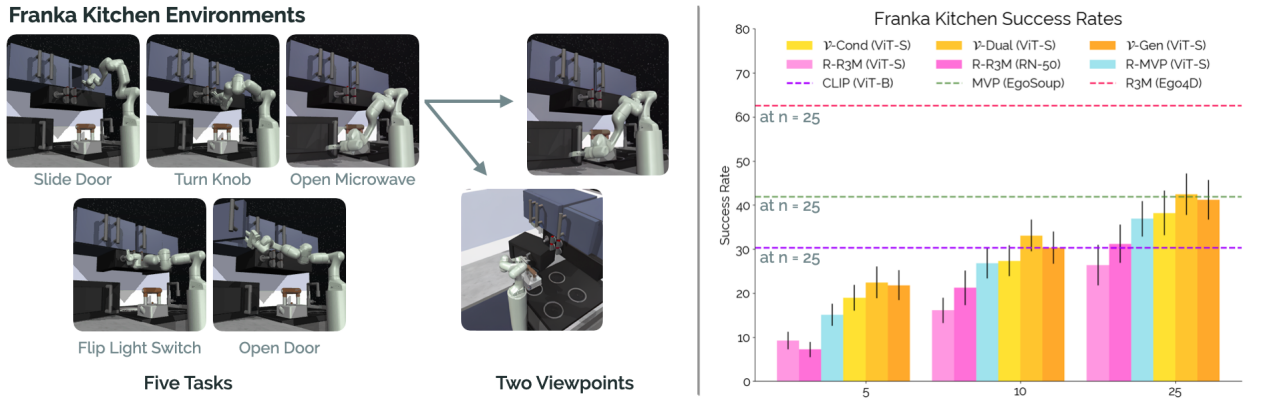

Figure 6. Franka Kitchen – Single-Task Visuomotor Control Results. Visualization of the Franka Kitchen evaluation environments, comprised of five unique tasks, with two camera viewpoints **[Left]**. Results (success rate for each of n demonstrations) for Voltron and baselines, showing the benefit of language-driven learning (over 3 seeds) **[Right]**. In dashed lines (not directly comparable), we plot *CLIP (ViT-B)*, *MVP (EgoSoup)*, and *R3M (Ego4D)* trained with $n = 25$ demonstrations.

Motivation. A large body of work looks at learning language-conditioned policies for human-robot collaborative settings (Arumugam et al. 2017; Stepputtis et al. 2020; Lynch and Sermanet 2020; Karamcheti et al. 2021b; Ahn et al. 2022). This evaluation probes the robustness and reliability of learned representations in a real-world setting.

Evaluation Details. We construct a "study desk" environment (Fig. 7) with five prototypical "tasks": 1) closing the drawer, 2) throwing the green bag of chips in the trash can, 3) discarding the used coffee pods, 4) moving the cyan coffee mug to the purple plate, and 5) moving the same mug to the yellow plate. For each task, we collect 20 teleoperated demonstrations at 10 Hz, randomly resetting the scene between episodes. We adopt the keyframe-based action space proposed in James and Davison (2022) for learning. This approach heuristically breaks a demonstration into 4-5 "waypoints" (end-effector poses) that are used as action targets during behavior cloning; during policy execution, we plan min-jerk trajectories from the current position to the predicted waypoint, feeding the subsequent state and visual observation back to our policy (James et al. 2022; Shridhar et al. 2022). To collect diverse instructions, we prompt ChatGPT (version dated Jan 9th, 2023; OpenAI 2022) with simple task descriptions, asking it to generate diverse language instructions, collecting 25 utterances total (20 train, 5 held-out) per task.[2] We parameterize our policy similarly to §5.3, adding a shallow MLP on top of the extracted (frozen) visual representations (Misra et al. 2017). This task is *language-conditioned*; as in OCID-Ref, we use the given language encoders for each approach where possible, appending DistilBERT features to pure visual representations otherwise. We report success rates with partial credit – 0.25 points for achieving each of the following "milestones:" reaching an object, interacting with it, transporting it, and completing the task. We include videos of policy rollouts on the project page.

Experimental Results. Looking at success rates of the various representations (Fig. 7; top right) we see an exaggerated version of the trends exhibited in the single-task control setting; Voltron models obtain an extra boost in performance across the board given that this task is language-conditioned, highlighting the strength of its fused representations. Similarly, R-R3M models exhibit the next best performance. Due to shared resource constraints, we do not run out *MVP (EgoSoup)*, *R3M (Ego4D)*, or *CLIP (ViT-B/16)*, though we expect similar trends.

5.5 Qualitative: Zero-Shot Intent Scoring

We perform a qualitative evaluation for the problem of language-based *intent scoring*; given a language expression describing an intent or behavior (e.g., "opening the faucet") and a corresponding video (that may or may not show the described behavior), predict an "alignment score" for each frame of a video. This alignment score should capture how well the current visual context matches the described behavior – ideally reflecting calibrated confidence over time (an example language/video is shown in Fig. 8; left).

Motivation. This evaluation is motivated by two active areas of research: reward learning from language and demonstrations (Smith et al. 2020; Shao et al. 2020; Chen et al. 2021a; Bahl et al. 2022), and belief modeling for human-robot collaboration (Hoffman and Breazeal 2007; Hauser 2012; Bandyopadhyay et al. 2013). This evaluation probes for the ability to reason over intents and visual behaviors *jointly, without the need for additional data or supervision*.

Evaluation Details. This is a qualitative evaluation that focuses on measuring how well existing approaches "track" progress conditioned on a language intent over time. Doing this zero-shot means that we can only evaluate models that can produce alignment scores given language and visual context: 1) *CLIP (ViT-B/16)* through cosine similarity of learned vision and text representations, 2) *R3M (Ego4D)* through the "video-language alignment" head, and 3) our **$\mathcal{V}$-Gen** model (by measuring the likelihood of a given language utterance conditioned on visual context under the language generator). Given a video of an agent performing some behavior described in language (e.g., "opening the faucet"), we estimate and plot scores under each model across a sequence of video frames. We use videos from WHiRL (Bahl et al. 2022) of humans and robots performing the same tasks from different views; we choose to evaluate intent scoring for both agents to better capture the robustness and transfer potential for these approaches in similar real-world settings.

Experimental Results. The two curves in Fig. 8 show the predicted scores over time for the language intent "opening

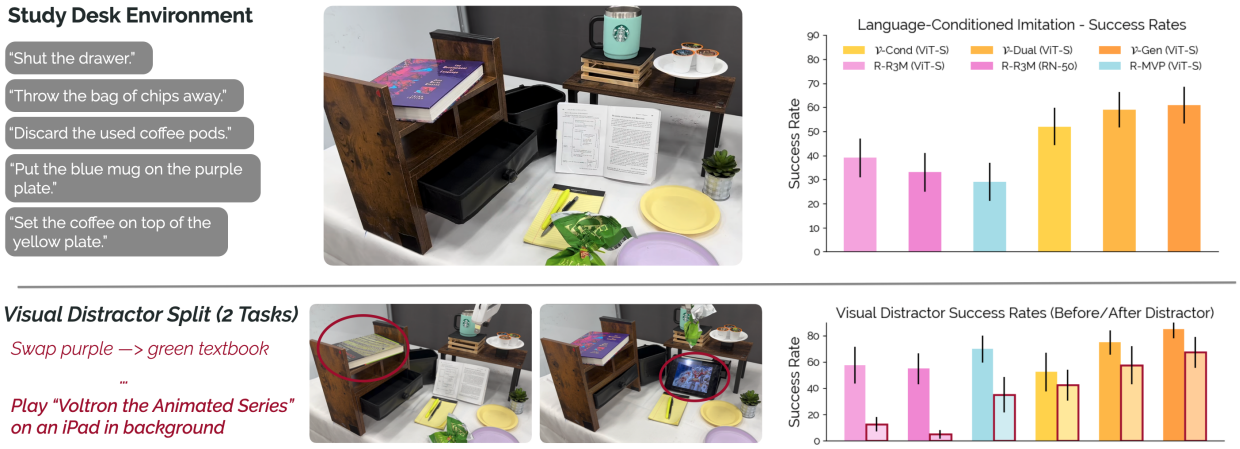

Figure 7. Real-World Language-Conditioned Imitation Learning Results. The real-world "Study Desk" environment, with sample language instructions corresponding to the five behaviors we evaluate. **[Top]** The challenging *visual distractor* split for evaluating robustness to novel distractors, ranging from simple color swapping of background objects (e.g., purple to green textbook), to more drastic changes such as playing a clip from *"Voltron – the Animated Series"* in the background **[Bottom]**.

the faucet." Even though it has never been trained for this task, we find that $\mathcal{V}$**-Gen** is able to coherently predict not only the exact frames corresponding to "keypoints" in each video (e.g., touching the handle, observing when the water starts running), but is also capable of measuring *partial progress* – akin to a shaped, dense reward; however, both *R3M (Ego4D)* and *CLIP (ViT-B/16)* fail at this task, predicting random scores with high variance across sequential time steps. Note that the intent scores are not perfect; after turning the faucet on for the human video, predicted scores remain high, while for the robot, the scores taper off. It is not clear why this happens, but given a small amount of adaptation data, one could ensure consistent behavior. We provide additional examples in §B.3.

6 Ablations & Further Analysis

The comparative results across the various evaluation problem domains paint Voltron's language-driven representations in a favorable light relative to MVP and R3M baselines. Yet, there remain key questions that we address in this section: is language supervision actually driving these results? Why generative language modeling over masked language modeling? How robust are Voltron representations?

Ablation: The Impact of Language Supervision. The second row of Table 4 shows a subset of evaluation results across three different problem domains when training a "no-language" variant of the $\mathcal{V}$**-Cond** architecture – this variant is in essence an alternate version of a masked autoencoder that uses the small architecture modifications we added for training stability in §4. As such, it also serves as an *architecture ablation* when compared to the R-MVP results, enabling us to isolate the impact of the small stability modifications described in §4. Indeed, the results confirm our hypotheses: first, removing language results in a definitive drop in performance across all evaluation applications. Second, the respective results for each evaluation application are on par with the corresponding results for the R-MVP model, demonstrating that the performance of Voltron models does not stem from the architecture.

Ablation: Generative vs. Masked Language Modeling. Looking at the Voltron objective, a natural question to ask is why we chose *language generation* over *masked language modeling*. This is especially relevant as concurrent work proposes learning multimodal masked autoencoders (M3AE) both within and outside of robotics (Geng et al. 2022; Liu et al. 2022), showing promising results in learning visual representations for image classification tasks, amongst others. To assess the differences, we choose to reproduce the M3AE model in a manner similar to our reproduction of MVP and R3M; we keep the same Something-Something-v2 pretraining data, adopting the exact procedure described in Geng et al. (2022), then evaluating the resulting representations on the same subset of evaluation domains as in the prior ablation (third row of Table 4). Surprisingly, we see drastic drops in performance *across the board*. Looking at the pretraining curves, we identify a possible reason for this failure: in optimizing M3AE on Sth-Sth, we see the language modeling loss go to zero almost *immediately*, leading to overfitting. A possible explanation is that the masked language modeling conditioned on visual contexts in datasets annotated with *short, predictable narrations* leads to degenerate representations, while generative language modeling is not susceptible to the same types of collapse.

Ablation: Scaling Model Size. Prior approaches show gains when scaling model size; here, we present evidence that Voltron models behave similarly. For each evaluation in §5, we evaluate a ViT-Base variant of $\mathcal{V}$**-Cond** (86M parameters vs. the 22M in the ViT-Small). We see universal improvement: Top-5% precision for grasping (Table 2; middle) increases by 15%, expression grounding accuracy improves (Table 3; middle), as does performance on control.

Table 4. Ablation Experiments. We select a subset of evaluations from §5 – grasp affordance prediction, referring expression grounding, and single-task visuomotor control.

	Grasp Top-1%	**Refer** Total %	**Imitate** (n = 25)
$\mathcal{V}$ + Lang [Ours]	80.71	89.38	38.2 ± 5.09
No-Language ↓	65.83	*53.44*	33.1 ± 4.79
R-M3AE ↓↓	*52.79*	*51.61*	*24.0 ± 4.21*

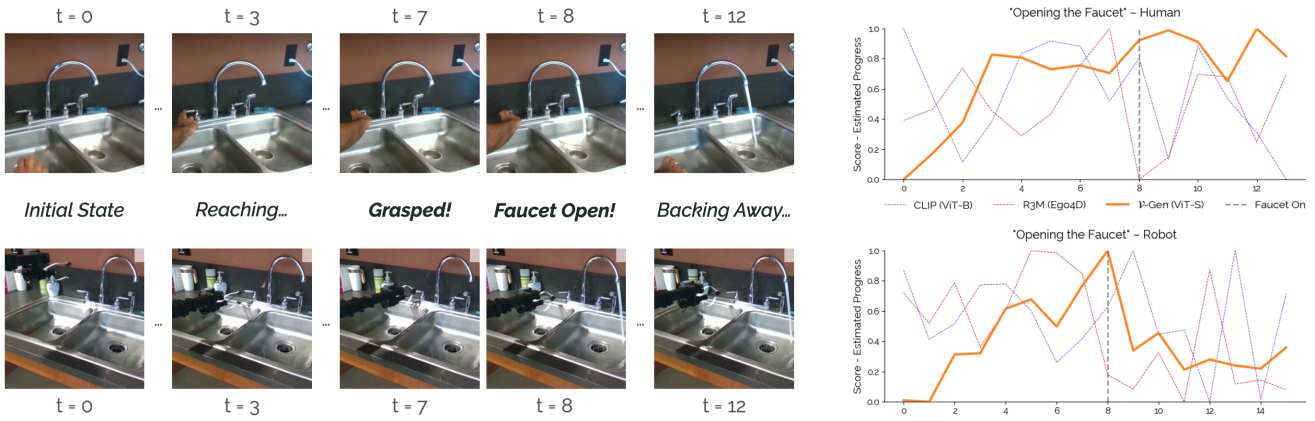

Figure 8. Qualitative Zero-Shot Intent Scoring Results. Given a pair of videos from the WHiRL dataset (Bahl et al. 2022) of a human and robot performing a task, we evaluate the ability of $\mathcal{V}$**-Gen**, R3M (from Nair et al. (2022)) and CLIP in scoring various frames subject to the utterance "opening the faucet." While CLIP and R3M produce extremely noisy scores, $\mathcal{V}$**-Gen** is *calibrated*, successfully tracking progress over time – both for the human user, *as well as for the robot*.

Analysis: Robustness to Real-World Distractors. Factors such as lighting conditions, time of day, and accidental environment perturbations (e.g., a colleague knocking over the camera) can have a profound impact on performance of robotic systems, especially if learned representations are not robust. We run a limited "robustness" evaluation after training language-conditioned policies from the demonstrations described in §5.4. Success rates before and after introducing visual distractors for two of the "meta-tasks" are in Fig. 7 (bottom right).[3] We find that Voltron and R-MVP models are robust to the most extreme distractors – a benefit of per-patch masking coupled with MAP-based extraction.

7 Capability-Aware Data Sourcing

The focus of this work so far has centered around Voltron as a general framework for learning flexible visual representations with language supervision. While our controlled experiments demonstrate the strengths of Voltron representations over alternative approaches, they miss a critical part of what makes a visual representation usable for an individual application – *data*. We introduce "capability-aware data sourcing," an intentional approach for building diverse pretraining mixtures that enable transfer to a wide range of robotics applications by explicitly considering the capabilities we want our representations to encode. We motivate this approach as follows: first, we present a negative result showing how naively scaling a pretraining dataset without designing around the given representation learning objective can lead to degraded representation quality. Specifically, we show that naively scaling on "in-the-wild" datasets of egocentric videos such as Ego4D (Grauman et al. 2022) are suboptimal for Voltron and similar approaches due to noise and artifacts present in the underlying videos. We then provide a review and extended discussion of alternative data sources for visual representation learning that capture a wealth of different capabilities. Finally, we use the lens of capabilities to assemble a compact pretraining mixture that enables transfer to a broad range of robotics applications. We demonstrate the efficacy of this dataset through a case study: we identify two distinct applications within robotics, each requiring representations that capture completely different capabilities and visual priors at multiple levels of abstraction. The remainder of the paper (§8) delves into the concrete implementation details for using this dataset to pretrain and adapt Voltron representations for each downstream application, with thorough evaluations and analysis.

7.1 Negative Result: Scaling Voltron on Ego4D

Existing representation learning approaches such as R3M and MVP leverage Ego4D (Grauman et al. 2022) as a source of diverse, egocentric videos that capture a wide range of objects and behaviors useful for robotics. Ego4D consists of over 3,000 hours (spanning millions of frames) of first-person egocentric videos, collected across 9 countries and 74 worldwide locations, coupled with *dense language narrations for the entirety of the dataset*. The raw diversity of this dataset coupled with its scale – 3-4 orders of magnitude larger than Sth-Sth (Goyal et al. 2017) – make it a promising target for scaling up Voltron pretraining. This potentially allows us to learn general-purpose representations that could generalize to a wide spectrum of tasks and environments.

Unfortunately, pretraining Voltron representations on Ego4D yields *worse representations* than training on Sth-Sth alone. Fig. 9 shows the evaluation performance across all tasks in our suite when training a $\mathcal{V}$**-Cond** model with a ViT Base backbone (86M parameters) on the original Sth-Sth representations (orange), and two subsets of the full Ego4D dataset – one consisting of 200K clips (comparable in size to Sth-Sth; light gray), and a much larger split of 1M clips (dark gray). For all Ego4D experiments, we fix the batch size and number of gradient steps to that of the base model trained on Sth-Sth, and keep all other hyperparameters constant. Yet, while training on the larger amount of frames from Ego4D-1M does lead to slight improvements over the Ego4D-200K split, the results in aggregate are *much worse* than training on Sth-Sth alone – a deficiency of over 40% for Grasping, 10% for referring expression grounding, and 5-10% on control.

To better understand this deficiency we ran a qualitative analysis, visualizing images from the dataset that exhibited high reconstruction loss *at the end of training*. We noticed two patterns that seemed particularly at odds with learning good representations – namely 1) a huge fraction of language narrations that mentioned objects or behaviors not present in a given image, or 2) a huge amount of visual artifacts such

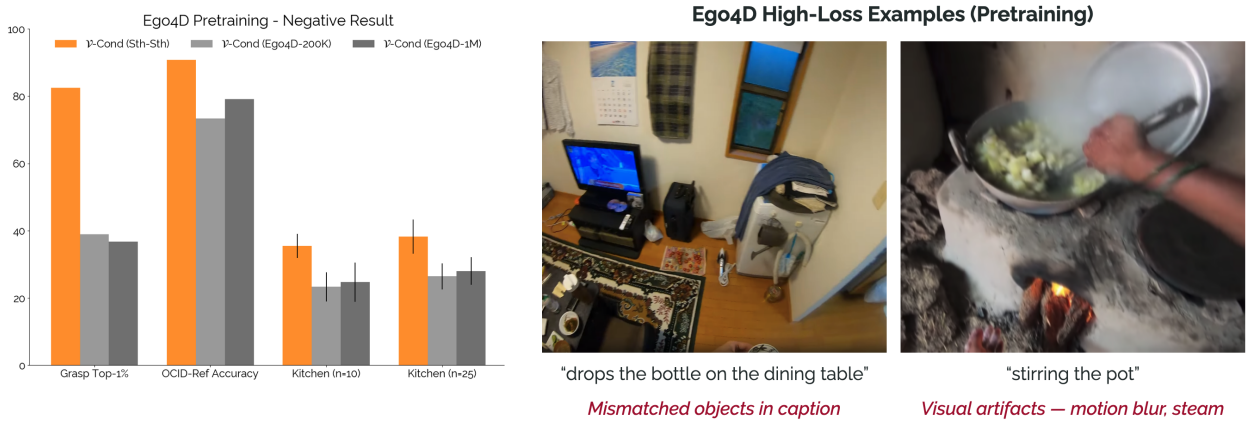

Figure 9. Ego4D Pretraining – Negative Results. We train on two splits of Ego4D (Grauman et al. 2022), one of 200K video clips (comparable in size to Sth-Sth), as well as a larger subset of 1M examples. Results shown are with the ViT-B/16 backbone, as compared to the same $\mathcal{V}$**-Cond** model trained *only* on Sth-Sth. On the right, we visualize high-loss examples from training, identifying issues such as mislabeled narrations and motion blur as detrimental to language-conditioned masked-autoencoder training.

as motion blur present in individual frames (see Fig. 9; right). These phenomena are particularly damaging for approaches like Voltron and general masked autoencoders that try to *reconstruct* individual pixels of an image from masked out contexts; fitting artifacts such as motion blur or aliasing with a per-pixel reconstruction objective is incredibly difficult, while learning to correlate perceptual features with completely unrelated language leads to spurious correlations at best (or completely learning to ignore language inputs at worst).

Concurrent work provides additional supporting evidence that "in-the-wild" datasets like Ego4D are suboptimal for representation learning approaches that rely on reconstruction objectives. Specifically, Dasari et al. (2023) find that training masked autoencoders on ImageNet frames, or on frames sourced from clean egocentric datasets (similar in nature to Sth-Sth) lead to stronger performance than training on Ego4D alone. Furthermore, both Dasari et al. (2023) and Majumdar et al. (2023) show that only after scaling beyond 1M frames (and ideally, to the 2M-5M frame range) do you start to see positive gains from training on noisy datasets such as Ego4D – a quantity of data that requires similarly expensive amounts of compute to fit. Instead, both our work and these concurrent investigations indicate that the most promising data sources are those that are *diverse* and *clean*, consisting of images spanning a variety of objects, behaviors, and scenes that ideally reflect the downstream applications and capabilities one cares about (Hansen et al. 2022; Burns et al. 2023), while remaining free of artifacts that hurt learning.

7.2 Alternative Data Sources for Visual Representation Learning

Recent and concurrent work present thousands of controlled experiments evaluating existing pretrained representations – and different choices of pretraining data – across applications and axes such as sample-efficiency, out-of-distribution generalization, robustness, and more (Hansen et al. 2022; Hu et al. 2023; Majumdar et al. 2023; Burns et al. 2023). Dasari et al. (2023) conclude that pretraining on *curated* datasets such as Kinetics (Carreira and Zisserman 2017) or 100 Days of Hands (Shan et al. 2020) leads to far better downstream performance than alternatives such as Ego4D, due to the cleanliness and diversity of the corresponding images. Further experiments show that pretraining reconstruction-based representations (using objectives similar to Voltron) on *mixtures* of clean datasets lead to even better downstream performance, even if the individual mixture components are relatively moderate in scale. Burns et al. (2023) present complimentary studies that examine the robustness and generalizability of existing representations under different types of distribution shift (e.g., lighting conditions, scene diversity, novel objects). One finding is that under distribution shifts, representations such as R3M and MVP pretrained on egocentric datasets like Ego4D perform *worse* than traditional representations pretrained for ImageNet classification. The second finding links the emergence of *segmenting features* – the ability of a representation learning backbone to learn implicit boundaries between objects, shapes, and materials – to downstream performance. Burns et al. (2023) cite models such as MoCov3 (Chen et al. 2021b) DINO (Caron et al. 2021), and DINOv2 (Oquab et al. 2023) that learn such features either as a result of their pretraining objective, or as a result of mixing pretraining datasets of different granularities. For example, DINOv2 trains on a mixture of datasets spanning ImageNet-22K, satellite images, and datasets for classification, segmentation, depth estimation, and retrieval capturing objects and scenes at multiple levels of abstraction. Finally, Majumdar et al. (2023) run over 10,000 GPU hours of pretraining ablations demonstrating the importance of 1) identifying (or preprocessing) pretraining data to maximize data cleanliness, and 2) pretraining on mixtures of different data sources that capture different granularities (from objects to tabletops to entire rooms).

These results pose a question: *what alternative data sources are actually useful for visual representation learning?* Taking both cleanliness (lack of visual artifacts, clear camera focus) and diversity as necessary conditions, we return to our thesis around "capability-aware data sourcing" and index different datasets by the downstream abilities they enable:

Robot Trajectories for Behavior Learning. One of the highest impact changes in the last year has been the release of large-scale, clean datasets of *robot trajectories* that span different camera viewpoints, scenes, and embodiments. One such effort is *Open-X Embodiment* (Padalkar et al. 2023) a collection over 60 individual datasets used in various

robotics works over the past decade, totaling over 1M+ robot trajectories and 22+ embodiments; much of the data is further annotated with language instructions and other semantic skill labels. Notably, Open-X Embodiment includes previously closed-source data such as the RT-1 and BC-Z (Brohan et al. 2023; Jang et al. 2021) datasets. For robot navigation datasets, *VINT* (Shah et al. 2023) provides a dataset of over 100 hours of real-world trajectories across 8 robot platforms and a variety of indoor and outdoor environments.

Curated Videos for Affordance Learning. While in-domain robot trajectories are a great source of learning, curated video datasets (either egocentric or allocentric) remain a strong source of behavior and affordance learning for long-tail object manipulation, navigation, and scene diversity. Many of the representation learning works identify *Something-Something-v2* (Sth-Sth; Goyal et al. 2017) as a strong pretraining dataset due to its cleanliness and diversity of objects and skills. As mentioned above, allocentric action-recognition datasets like *Kinetics* (Carreira and Zisserman 2017), *HowTo100M* (Miech et al. 2019), and *Epic-Kitchens* (Damen et al. 2018) also provide cleanliness and diversity in terms of scenes and objects. For indoor navigation, egocentric datasets such as *Real Estate 10K* (Zhou et al. 2018) and *Open House 24* are large and tested (Majumdar et al. 2023).

Broad-Coverage Image Datasets. ImageNet (Deng et al. 2009) remains a strong and diverse source of image data for learning general priors about shapes, objects, and scenes. Other curated datasets that capture a rich distribution of scenes and objects include *MS COCO* (Lin et al. 2014) and *Visual Genome* (Krishna et al. 2017). Common object detection datasets such as *LVIS* (Gupta et al. 2019), *Objects365* (Shao et al. 2019) cover a broad spectrum of scenes and object categories. A wealth of follow-up works provide additional annotations like natural language referring expressions, scene graph annotations, or visual queries – examples include *RefCOCO* (Yu et al. 2016) and *VQA* (Agrawal et al. 2015).

Granular Image Data for Visual Reasoning. One of the key conclusions from prior work is the importance of capturing a mixture of granularities when building a pretraining dataset (Dasari et al. 2023; Burns et al. 2023); especially for robotic manipulation and navigation, understanding the differences between a scene, an object, and individual parts of an object are important for learning affordances and general reasoning about the visual world. While broad-coverage image datasets like COCO capture high-level objects and relationships, they miss more granular features around parts of objects ("handles," "knobs") and affordances thereof. Prior work use datasets such as *100 Days of Hands* (Shan et al. 2020) and *RoboNet* (Dasari et al. 2019) to capture some of this granularity. Other, more recently introduced datasets that are explicitly annotated to capture various granularities include datasets such as *EgoObjects* (Zhu et al. 2023) and *Parts and Attributes of Common Objects* (PACO; Ramanathan et al. 2023). These datasets in particular provide an additional benefit, in that they are sourced from Ego4D (Grauman et al. 2022) with special care to ensure clean data; they thus reflect both the egocentric viewpoint useful for robotics applications, as well as the raw diversity of objects and scenes present in Ego4D.

Summary. These datasets form a solid basis for learning representations that can be used for myriad robotics applications. They are also *compact*; compared to the millions of frames in datasets such as Ego4D, these datasets are small, consisting of 40K - 200K examples on average, enabling one to craft lightweight mixtures for learning.

7.3 Case Study: Sourcing a Dataset for Control and Open-Vocabulary Detection

The remainder of this work grounds capability-aware data sourcing and Voltron as a flexible representation learning framework in a concrete example. Our goal is to show how one might use these two ideas to efficiently learn powerful representations that are suitable for a wide range of robotics applications. To do this, we present a case study for developing end-to-end systems built on top of Voltron representations for two distinct applications. The first application focuses on visuomotor policy learning – specifically, learning dense state representations that enable sample-efficient policy learning as evaluated in the Franka Kitchen evaluations in §5.3. The second application focuses on open-vocabulary part and object detection – for example, predicting bounding boxes for all instances of "metal handles" or "scissors with red handles" in a given scene.

Taken together, these applications capture the diverse range of visual priors and capabilities a pretrained representations must encode in order to be broadly useful for robotics. Visuomotor policy learning requires representations that capture high-level features around behaviors, affordances, and scene-level semantics that transfer across embodiments. Separately, fine-grained open-vocabulary detection requires grounded visual representations that are granular, picking up on features at the level of individual parts, attributes, and material composition of objects and object interactions. We can explicitly tie these capabilities to the prescriptive data sources indexed in §7.2 to inform our pretraining mixture: 1) *Open-X Embodiment* (Padalkar et al. 2023) to capture generalization across different (robot) embodiments and common behaviors and affordances, 2) *Something-Something-v2* (Sth-Sth; Goyal et al. 2017) to capture a more diverse range of behaviors and scenes, 3) *MS COCO* (Lin et al. 2014) to learn grounded scene and object representations, and 4) *Parts and Attributes of Common Objects* (PACO; Ramanathan et al. 2023) to learn finer-grained part and attribute features.

The corresponding dataset – referred to as the Voltron-X pretraining dataset – consists of 130,271 robot demonstrations from Open-X Embodiment, 168,911 video clips from Sth-Sth with associated language annotations, 118,287 images with corresponding captions from MS COCO (using the 2017 version of the dataset), and 61,457 images from PACO (sourced from two separate domains – LVIS, and Ego4D). We specifically use the "clean" split of Open-X Embodiment prescibed by recent work (Ghosh et al. 2023), consisting of the RT-1 Robot Action dataset (Brohan et al. 2023), the Bridge-v2 dataset (Walke et al. 2023), the Freiburg Franka Play dataset (Rosete-Beas et al. 2022; Mees et al. 2023), the CMU Franka Pick-Insert dataset (Saxena et al. 2023), and the UT Austin Tabletop datasets (Nasiriany et al. 2022; Liu et al. 2023a). For all datasets, we only use data from the predefined "train" split if specified, leaving any validation or test data untouched. The aggregate pretraining dataset is compact, consisting of 478,926 examples – only three times the size of the Sth-Sth

dataset used to train the Voltron representations in the earlier part of this work (§4, §5). More importantly, this mixture is half the size of ImageNet, and *one tenth* the size of the datasets used to train R3M, MVP, and VC-1 (Nair et al. 2022; Radosavovic et al. 2022; Majumdar et al. 2023).

8 Voltron-X: Adapting Representations from Capability-Aware Data

In the following sections we step through using the Voltron-X pretraining mixture defined in §7.3 to adapt representations for different downstream applications. We first provide implementation details for base representation learning – we refer to these new representations as Voltron-X ($\mathcal{VX}$) representations to distinguish them from the models trained in §4. We then provide details and evaluation experiments around adapting $\mathcal{VX}$ representations for each of the two applications in our case study.

8.1 Learning Voltron-X Representations

Pretraining $\mathcal{VX}$ representations mirrors the process described in Sec. §3. As only a subset of the dataset consists of videos, we focus primarily on $\mathcal{V}$**-Cond** representations, learning to reconstruct individual frames conditioned on language input. While Sth-Sth and MS COCO are fully annotated with language narrations and captions respectively, the Open-X Embodiment dataset is only partially annotated with language annotations. Similarly, the images in the PACO dataset are only labeled with federated object, part, and attribute labels. In order to train $\mathcal{V}$**-Cond** representations on non-annotated demonstrations from the Open-X Embodiment dataset, we simply add a "null" language input consisting of just the `<CLS>` token used to denote the beginning of a sentence for BERT models (Devlin et al. 2019). In order to train on images from the PACO dataset, we map the set of annotated objects and parts present in a given image to a list of natural language descriptors using `text-ada-001`, the smallest GPT-3 model provided by OpenAI. As an example, `text-ada-001` maps the official symbolic part label "bottle:cap" to the natural language string "bottle cap." We perform a light filtering pass through all the data, removing any videos with fewer than 5 frames, and any images with ultra-wide or ultra-tall aspect ratios (2:1, 1:2) in order to prevent warping.

We train $\mathcal{V}$**-Cond** using the ViT-Base architecture (86M parameters), following the procedure outlined in §4. We train for 100 epochs with a batch size of 1024, *without* any dropout or data augmentation. For these experiments, we use GPU compute provided by AWS, with a total train time of approximately 23 hours across 8 GPUs.

8.2 Time-Contrastive Adaptation for Visuomotor Control

In this section, we show how we can use simple learning objectives to lightly adapt $\mathcal{VX}$ representations for sample-efficient visuomotor policy learning. By themselves, Voltron representations produce localized features for individual patches in an image. However, for control applications, prior work (Parisi et al. 2022) finds that *dense state features* are most amenable to policy learning from limited demonstrations.

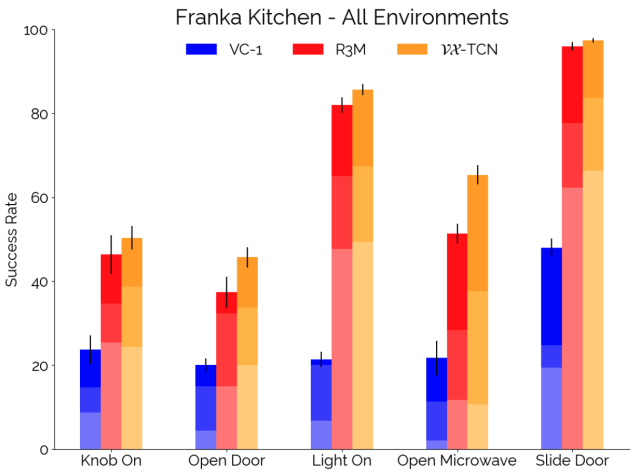

Figure 10. Franka Kitchen Imitation Learning Results. We plot success rate (across 2 camera viewpoints, 3 random seeds) for VC-1 (Large) (Majumdar et al. 2023), R3M (Nair et al. 2022), and $\mathcal{VX}$-TCN, for each of the five Franka Kitchen tasks. The stacked bars correspond to few-shot success rates for different numbers of demonstrations in $n \in 5, 10, 25$.

As a result, we need an objective for distilling the existing patch features into a global state representation.

Inspired by standard approaches for learning global state features such as R3M, VIP, and LIV (Nair et al. 2022; Ma et al. 2022, 2023), we adopt *time-contrastive learning* (TCN; Sermanet et al. 2018) as our adaptation objective. The TCN objective is simple and general to any dataset of videos. Given a learned frame encoder $F_\theta(o_i) \in \mathbb{R}^k$ and proper similarity metric $\mathcal{S}$ (in this work, we take $\mathcal{S}$ to be negative Euclidean distance), one samples a triple of frames from a given video (o_i, o_j, o_k) such that $i < j < k$, $(z_i = F_\theta(o_i), z_j = F_\theta(o_j), z_k = F_\theta(o_k)$ and minimizes the following contrastive loss:

$$\mathcal{L}(\theta) = -\log\left(\frac{\exp\left[\mathcal{S}(z_i, z_j)\right]}{\exp\left[\mathcal{S}(z_i, z_j)\right] + \exp\left[\mathcal{S}(z_i, z_k)\right]}\right)$$

This objective encourages the learned encoder F_θ to group nearby states (observations close in time) together, while pushing any other state further away. As a result, the learned global state features are strongly biased towards encoding temporal dynamics, capturing the high-level semantics around individual behaviors or object interactions.

We implement our TCN encoder F_θ on top of our pretrained $\mathcal{VX}$ representations by initializing a 2-Layer Multiheaded Attention Pooling (MAP) extractor (Lee et al. 2018), identical to the MAP blocks used in our prior evaluations (§5.1, §5.2). The MAP extractor is a simplified Transformer encoder that takes as input the sequence of patch embeddings output by the $\mathcal{VX}$ backbone, and outputs a single learned pooling vector that we take to be our TCN representation z. To facilitate efficient training, we *freeze* the $\mathcal{VX}$ backbone, only updating the weights of the MAP extractor during TCN adaptation. We note that this formulation of the TCN objective is much simpler than the R3M objective, which additionally adds auxiliary losses for video-language alignment, and representation sparsity; adding a separate loss for language alignment is unnecessary given the multimodal nature of the base $\mathcal{VX}$ representations, and we find that a separate representation sparsity penalty is

redundant when combined with the LayerNorm layers used by default when training Transformer neural networks.

We run TCN adaptation for 100K gradient steps (approximately 3 hours on 8 GPUs) on the combined Sth-Sth and Open-X Embodiment datasets from §8.1. In order to learn representations that are language-agnostic, we drop out language conditioning for each example with $p = 0.5$. We adopt the remaining hyperparameters from R3M, training with a learning rate of 1e-4 and a local batch size of 32. The resulting TCN-adapted model $\mathcal{VX}$-TCN takes in individual visual observations (e.g., from a camera mounted on a robot base), and produces global state features with $d = 768$.

Evaluation Results. To evaluate the adapted $\mathcal{VX}$-TCN representations for sample-efficient visuomotor policy learning, we use the few-shot Franka Kitchen control suite introduced in §5.3. Given different size demonstration datasets with $n = 5, 10, 25$, we aim to learn a shallow MLP policy for producing 9-DoF joint actions given the current visual state. Notably, whereas the evaluation earlier in the paper adapts Voltron patch representations with expressive policy heads, here we perform a compute-constrained evaluation where we only train a 2-layer MLP with hidden dimension $d = 256$ on top of the *frozen* global state representations. This directly evaluates how well $\mathcal{VX}$-TCN encodes features useful for control.

Results on all five Franka Kitchen tasks (averaged over 2 unique camera viewpoints and 3 random trials) are presented in Fig. 10. As baselines, we compare $\mathcal{VX}$-TCN representations to the publicly released R3M representations trained on the entirety of Ego4D, and the 307M parameter VC-1 Large representations (Majumdar et al. 2023) trained on the union of Ego4D, ImageNet, and various navigation datasets (a pretraining dataset comprised of over 5.6M frames). We find that on almost all tasks, at each demonstration dataset size, $\mathcal{VX}$-TCN significantly outperform both R3M and VC-1 representations. On two tasks ("Knob On" and "Open Microwave"), $\mathcal{VX}$-TCN representations slightly underperform R3M representations at $n = 5$ demonstrations, but quickly equalize when training on the full dataset.

These results show the clear wins in compute and data efficiency afforded by the combination of Voltron-X representations and the capability-aware dataset sourcing. With a dataset comprised of less than 500K examples and only 10s of GPU hours, the $\mathcal{VX}$-TCN representations are able to outperform R3M and VC-1 representations trained on orders of magnitude more data and compute. To make this concrete – R3M trains for 1.5 million steps on Ego4D, while VC-1 trains for 182 epochs over 5.6 million frames – on the order of hundreds of GPU hours and 10x the data.

8.3 DETR Adaptation for Open-Vocabulary Part & Object Detection

The second application we consider in our case study is that of open-vocabulary part and object detection; given a referring expression or phrase in natural language (e.g., "metal handles," or "the blue coffee mug above the sink") output a (possibly empty) set of bounding boxes and corresponding confidence scores for *all instances of the given referent in the scene*. Focusing on this application serves an addition purpose to developing stronger representations for robotics – that of *need*. As a community, we lack open-vocabulary detectors that are specifically suited to robotics applications, where we want to detect granular parts of objects from free-form queries (e.g., handles made of specific materials, or objects of certain shapes or colors). Existing pretrained models for closed set detection such as YOLOv7 (Wang et al. 2022) or ViTDet (Li et al. 2022) are limited by the fixed set of categories they are trained on, demonstrating a lack of flexibility; similarly, existing open-vocabulary detectors such as OWL-ViT (Minderer et al. 2022, 2023) are limited by the pretrained vision-language representations they train with (e.g., CLIP), demonstrating a lack of sensitivity to fine-grained visual features. Especially as open-vocabulary detectors grow in use as a component in modular robotic systems (Zeng et al. 2023; Wu et al. 2023; Huang et al. 2023), developing open-vocabulary detectors that can handle different levels of granularity will be critical.

Adapting $\mathcal{VX}$ representations for open-vocabulary part and object detection is simple and straightforward, unlike other approaches for object detection that require multiple phases of training (Liu et al. 2023b), or hand-designed heuristics (Jocher et al. 2020; Wang et al. 2022). We adopt a single-stage training approach inspired by DETR and MDETR (Carion et al. 2020; Kamath et al. 2021). Given an image and corresponding language query (referring expression, or phrase to ground in the scene), we predict a fixed number K detection proposals, each associated with a corresponding "detection embedding" $r \in \mathbb{R}^d$. Similarly, we encode the language phrase into a "phrase embedding" $u \in \mathbb{R}^d$ of the same dimensionality as the individual detection embeddings. We compute a confidence score for each detection (e.g., does the given detection actually match the given phrase) by taking the dot product of each detection embedding r and phrase embedding u. Finally, we learn a shallow MLP to predict bounding box coordinates (x, y, w, h) from each detection embedding r. The objective for training the open-vocabulary detector is then the sum of three different losses – the binary cross-entropy loss on the predicted confidences (per detection), and the standard L1 and GIoU losses between the predicted and ground-truth bounding boxes (Rezatofighi et al. 2019).

We train this model on the remaining image datasets from §8 – MS COCO and PACO. To enable our model to ground full on referring expressions (e.g., "the black cup on the right of the sink") we use the RefCOCO annotations (Yu et al. 2016), consisting of crowdsourced language expressions mapped to individual bounding boxes for a subset of 28,158 images from the COCO dataset. To detect parts and objects from natural language queries, we adopt the same protocol for mapping the symbolic instance categories and attributes in PACO to individual language expressions, using OpenAI's `text-ada-001`. Similar to $\mathcal{VX}$-TCN adaptation, we run DETR-style adaptation for 100K gradient steps (approximately 4 hours on 8 GPUs). As each image in the PACO datasets can be annotated with multiple categories (object, part, or attribute), we uniformly sample from the full set each time we see an example. In order to handle cases where language phrases refer to objects that *do not exist in the given image*, we sample "negative categories" for an image with probability $p = 0.1$. We train with a learning rate of 2e-4, with a batch size of 512, and weight decay of 0.01.

Figure 11. Voltron-X Open-Vocabulary Grounding Visualizations. We visualize the predicted detections (bounding boxes) output by our open-vocabulary detector for various scenes and queries, including a set of"in-the-wild" robot images. Our model is able to ground simple phrases and full referring expressions to corresponding instances in each image. However, despite pretraining on robot images, detection performance on robot images remains high-variance, demonstrating a key limitation of our finetuning datasets.

Evaluation Results. We evaluate $\mathcal{VX}$-Detect model on the PACO Query challenge set, using the official evaluation metrics. The PACO Query challenge set consists of a series of natural language queries grouped into three levels (L1, L2, L3) of increasingly complexity across both the LVIS and Ego4D domains covered by the PACO dataset. L1 Queries are made up of single object/part categories ("scissors," "guitar body"), L2 Queries compose 2 attributes and categories ("blue striped mug") and L3 Queries are more complex ("a blue striped mug with a white handle"). Each query is associated with a single positive image where the given referent appears, and up to 100 negative images; the official evaluation reports Average Recall @ 5, where the average is taken over different IoU thresholds, following standard practice.

We report results on both the LVIS and Ego4D split, broken down by query in Table 5. We compare the performance of our lightly adapted $\mathcal{VX}$-Detect models to closed-set ViT-Det models (Li et al. 2022) of various sizes trained on PACO, as well as OWL-VIT (Minderer et al. 2022), and open-vocabulary object detection model that is adapted from CLIP (Radford et al. 2021) and trained on multiple standard object detection datasets. $\mathcal{VX}$-Detect outperforms both the closed-set and open-vocabulary detectors across both domains, on almost all query levels – with performance especially strong as queries grow in complexity, resembling full-blown referring expressions as opposed to individual category labels. These results again demonstrate the wins afforded by flexible Voltron-X representations and capability-aware dataset sourcing. The ability to encode both language and visual features within the same model enables flexibility to different types of language queries, while the focus on identifying datasets that capture low-level part and object detection enable granular feature learning.

Fig. 11 provides additional qualitative results. The top row depicts performance on a series of held out queries from the PACO LVIS split; not only is the model able to predict boxes for multiple object instances in a scene, but it shows an able to reason over compositions of objects, parts, and attributes. However, the model is not perfect – as evidenced by the middle example, it sometimes is too confident in its predictions, possibly due to the low negative sampling rate during training. More relevant are the results in the bottom row of Fig. 11, showing the performance of $\mathcal{VX}$-Detect on *"in-the-wild" images of robots* (Bahl et al. 2023). While the detection results are impressive having never seen images of robots during the adaptation phase, we see a clear limitation

Table 5. Results on PACO Zero-Shot Query Detection. We report Average Recall @ 5 for both the LVIS and Ego4D splits of the Parts and Attributes of Common Objects (PACO) Dataset, following the official evaluation protocol (Ramanathan et al. 2023). The $\mathcal{VX}$-Detect model sees impressive performance across the board, especially on L2 and L3 queries stemming from its ability to handle nuanced referring expressions.

	PACO – LVIS			PACO – Ego4D		
	L1 Queries	L2 Queries	L3 Queries	L1 Queries	L2 Queries	L3 Queries
ViT-Det FPN (ViT-B)	49.5	44.9	45.7	24.4	19.5	18.1
ViT-Det FPN (ViT-L)	**60.8**	55.6	59.0	36.9	33.3	34.9
OWL-ViT (CLIP ViT-B)	54.9	52.3	53.6	37.7	32.5	27.2
$\mathcal{VX}$-Detect (ViT-B)	40.1	**61.7**	**77.9**	**38.6**	**57.1**	**60.4**

in the current model's ability to detect objects in the presence of robot occlusions; as evidenced by the last example, as the "out-of-distribution" robot takes up more of the frame, the $\mathcal{VX}$-Detect model is more likely to be overconfident.

9 Conclusion

We propose Voltron, a framework for language-driven representation learning that balances *conditioning* and *generation* to learn flexible features that capture both low and high-level visual features. We introduce an evaluation suite of diverse problems within robotics for holistically evaluating visual representations, and perform a series of controlled data experiments and ablations that validate the strengths of our framework. We then motivate *capability-aware dataset sourcing* to identify a compact mixture of pretraining datasets that enable transfer to a range of robotics applications.

Using this mixture, we pretrain and adapt a new set of representations – Voltron-X – that outperform the existing state-of-the-art in two distinct applications. On sample-efficient policy learning, we produce better representations for control than R3M and VC-1 – approaches trained on orders of magnitude more data and compute. For open-vocabulary part and object detection, we outperform both closed-set detectors trained specifically for granular detection, as well as "generalist" open-vocabulary detectors such as OWL-ViT. Put together, this work highlights Voltron as a a flexible framework for learning visual representations for robotics.

Notes

1. Training Repository (with Pretrained Models): https://github.com/siddk/voltron-robotics. Evaluation Repository: https://github.com/siddk/voltron-evaluation Project Page: https://sites.google.com/view/voltron-robotics.
2. ChatGPT Prompt: *I'm trying to train a robot assistant that can follow diverse language instructions. One task requires moving an empty chip bag (a green bag of those jalapeno chips) to the garbage. Can you generate 25 natural-sounding instructions (e.g., "throw away the chips")?*
3. We try five distractors spanning simple changes such as swapping the purple textbook for a green one, to more extreme distractors such as playing clips *"Voltron, the Animated Series"* on a tablet in the middle of the workspace.

Acknowledgements

This work would not have been possible without the support of entire communities of students, engineers, and various domain experts; our gratitude cannot be understated. We would specifically like to thank Shyamal Buch, David Hall, and John Thickstun for their invaluable advice and suggestions around pretraining and evaluation. We further thank Dilip Arumugam, Ashwin Balakrishna, Suneel Belkhale, Masha Itkina, Tyler Lum, Vivek Myers, Karl Pertsch, and Blake Wulfe for their feedback on earlier drafts. Finally, we thank our RSS reviewers for their feedback on the original conference version of this work.

References

Aghajanyan A, Huang B, Ross C, Karpukhin V, Xu H, Goyal N, Okhonko D, Joshi M, Ghosh G, Lewis M and Zettlemoyer L (2022) CM3: A causal masked multimodal model of the internet. *arXiv preprint arXiv:2201.07520* .

Agrawal A, Lu J, Antol S, Mitchell M, Zitnick CL, Parikh D and Batra D (2015) VQA: Visual question answering. *International Journal of Computer Vision* 123: 4–31.

Ahn M, Brohan A, Brown N, Chebotar Y, Cortes O, David B, Finn C, Gopalakrishnan K, Hausman K, Herzog A, Ho D, Hsu J, Ibarz J, Ichter B, Irpan A, Jang E, Ruano RJ, Jeffrey K, Jesmonth S, Joshi NJ, Julian RC, Kalashnikov D, Kuang Y, Lee KH, Levine S, Lu Y, Luu L, Parada C, Pastor P, Quiambao J, Rao K, Rettinghouse J, Reyes DM, Sermanet P, Sievers N, Tan C, Toshev A, Vanhoucke V, Xia F, Xiao T, Xu P, Xu S and Yan M (2022) Do as I can, not as I say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691* .

Alayrac JB, Donahue J, Luc P, Miech A, Barr I, Hasson Y, Lenc K, Mensch A, Millican K, Reynolds M, Ring R, Rutherford E, Cabi S, Han T, Gong Z, Samangooei S, Monteiro M, Menick J, Borgeaud S, Brock A, Nematzadeh A, Sharifzadeh S, Binkowski M, Barreira R, Vinyals O, Zisserman A and Simonyan K (2022) Flamingo: a visual language model for few-shot learning. In: *Advances in Neural Information Processing Systems (NeurIPS)*.

Arumugam D, Karamcheti S, Gopalan N, Wong LLS and Tellex S (2017) Accurately and efficiently interpreting human-robot instructions of varying granularities. In: *Robotics: Science and Systems (RSS)*.

Bahl S, Gupta A and Pathak D (2022) Human-to-robot imitation in the wild. In: *Robotics: Science and Systems (RSS)*.

Bahl S, Mendonca R, Chen L, Jain U and Pathak D (2023) Affordances from human videos as a versatile representation for robotics. In: *Computer Vision and Pattern Recognition (CVPR)*.

Bandyopadhyay T, Won KS, Frazzoli E, Hsu D, Lee WS and Rus D (2013) Intention-aware motion planning. In: *Workshop for the Algorithmic Foundations of Robotics (WAFR)*.

Bao H, Dong L and Wei F (2022) BEiT: BERT pre-training of image transformers. In: *International Conference on Learning Representations (ICLR)*.

Bohg J, Morales A, Asfour T and Kragic D (2013) Data-driven grasp synthesis—a survey. *IEEE Transactions on Robotics (T-RO)* 30: 289–309.

Bommasani R, Hudson DA, Adeli E, Altman R, Arora S, von Arx S, Bernstein MS, Bohg J, Bosselut A, Brunskill E, Brynjolfsson E, Buch S, Card D, Castellon R, Chatterji N, Chen A, Creel K, Davis JQ, Demszky D, Donahue C, Doumbouya M, Durmus E, Ermon S, Etchemendy J, Ethayarajh K, Fei-Fei L, Finn C, Gale T, Gillespie L, Goel K, Goodman N, Grossman S, Guha N, Hashimoto T, Henderson P, Hewitt J, Ho DE, Hong J, Hsu K, Huang J, Icard T, Jain S, Jurafsky D, Kalluri P, Karamcheti S, Keeling G, Khani F, Khattab O, Koh PW, Krass M, Krishna R, Kuditipudi R, Kumar A, Ladhak F, Lee M, Lee T, Leskovec J, Levent I, Li XL, Li X, Ma T, Malik A, Manning CD, Mirchandani S, Mitchell E, Munyikwa Z, Nair S, Narayan A, Narayanan D, Newman B, Nie A, Niebles JC, Nilforoshan H, Nyarko J, Ogut G, Orr L, Papadimitriou I, Park JS, Piech C, Portelance E, Potts C, Raghunathan A, Reich R, Ren H, Rong F, Roohani Y, Ruiz C, Ryan J, Ré C, Sadigh D, Sagawa S, Santhanam K, Shih A, Srinivasan K, Tamkin A, Taori R, Thomas AW, Tramèr F, Wang RE, Wang W, Wu B, Wu J, Wu

Y, Xie SM, Yasunaga M, You J, Zaharia M, Zhang M, Zhang T, Zhang X, Zhang Y, Zheng L, Zhou K and Liang P (2021) On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258* .

Brohan A, Brown N, Carbajal J, Chebotar Y, Dabis J, Finn C, Gopalakrishnan K, Hausman K, Herzog A, Hsu J, Ibarz J, Ichter B, Irpan A, Jackson T, Jesmonth S, Joshi NJ, Julian RC, Kalashnikov D, Kuang Y, Leal I, Lee KH, Levine S, Lu Y, Malla U, Manjunath D, Mordatch I, Nachum O, Parada C, Peralta J, Perez E, Pertsch K, Quiambao J, Rao K, Ryoo MS, Salazar G, Sanketi PR, Sayed K, Singh J, Sontakke SA, Stone A, Tan C, Tran H, Vanhoucke V, Vega S, Vuong QH, Xia F, Xiao T, Xu P, Xu S, Yu T and Zitkovich B (2023) RT-1: Robotics transformer for real-world control at scale. In: *Robotics: Science and Systems (RSS)*.

Burns K, Witzel Z, Hamid JI, Yu T, Finn C and Hausman K (2023) What makes pre-trained visual representations successful for robust manipulation? *arXiv preprint arXiv:2312.12444* .

Carion N, Massa F, Synnaeve G, Usunier N, Kirillov A and Zagoruyko S (2020) End-to-end object detection with transformers. In: *European Conference on Computer Vision (ECCV)*.

Caron M, Touvron H, Misra I, Jegou H, Mairal J, Bojanowski P and Joulin A (2021) Emerging properties in self-supervised vision transformers. In: *International Conference on Computer Vision (ICCV)*.

Carreira J and Zisserman A (2017) Quo vadis, action recognition? a new model and the kinetics dataset. In: *Computer Vision and Pattern Recognition (CVPR)*.

Chaumette F and Hutchinson SA (2006) Visual servo control. I. basic approaches. *IEEE Robotics & Automation Magazine* 13: 82–90.

Chen AS, Nair S and Finn C (2021a) Learning generalizable robotic reward functions from "in-the-wild" human videos. In: *Robotics: Science and Systems (RSS)*.

Chen X, Xie S and He K (2021b) An empirical study of training self-supervised vision transformers. In: *International Conference on Computer Vision (ICCV)*.

Chowdhery A, Narang S, Devlin J, Bosma M, Mishra G, Roberts A, Barham P, Chung HW, Sutton C, Gehrmann S, Schuh P, Shi K, Tsvyashchenko S, Maynez J, Rao A, Barnes P, Tay Y, Shazeer NM, Prabhakaran V, Reif E, Du N, Hutchinson B, Pope R, Bradbury J, Austin J, Isard M, Gur-Ari G, Yin P, Duke T, Levskaya A, Ghemawat S, Dev S, Michalewski H, García X, Misra V, Robinson K, Fedus L, Zhou D, Ippolito D, Luan D, Lim H, Zoph B, Spiridonov A, Sepassi R, Dohan D, Agrawal S, Omernick M, Dai AM, Pillai TS, Pellat M, Lewkowycz A, Moreira E, Child R, Polozov O, Lee K, Zhou Z, Wang X, Saeta B, Diaz M, Firat O, Catasta M, Wei J, Meier-Hellstern K, Eck D, Dean J, Petrov S and Fiedel N (2022) PaLM: Scaling language modeling with pathways. *arXiv* .

Correll N, Bekris KE, Berenson D, Brock O, Causo AJ, Hauser KK, Okada K, Rodriguez A, Romano JM and Wurman PR (2016) Analysis and observations from the first amazon picking challenge. *Science* 15: 172–188.

Cui Y, Niekum S, Gupta A, Kumar V and Rajeswaran A (2022) Can foundation models perform zero-shot task specification for robot manipulation? In: *Learning for Dynamics & Control Conference (L4DC)*.

Damen D, Doughty H, Farinella GM, Fidler S, Furnari A, Kazakos E, Moltisanti D, Munro J, Perrett T, Price W and Wray M (2018) Scaling egocentric vision: The EPIC-KITCHENS dataset. In: *European Conference on Computer Vision (ECCV)*.

Dasari S, Ebert F, Tian S, Nair S, Bucher B, Schmeckpeper K, Singh S, Levine S and Finn C (2019) Robonet: Large-scale multi-robot learning. In: *Conference on Robot Learning (CoRL)*.

Dasari S, Srirama MK, Jain U and Gupta A (2023) An unbiased look at datasets for visuo-motor pre-training. In: *Conference on Robot Learning (CoRL)*.

Deng J, Dong W, Socher R, Li LJ, Li K and Fei-Fei L (2009) ImageNet: A large-scale hierarchical image database. In: *Computer Vision and Pattern Recognition (CVPR)*. pp. 248–255.

Devlin J, Chang MW, Lee K and Toutanova K (2019) BERT: Pre-training of deep bidirectional transformers for language understanding. In: *Association for Computational Linguistics (ACL)*. pp. 4171–4186.

Gelada C, Kumar S, Buckman J, Nachum O and Bellemare MG (2019) Deepmdp: Learning continuous latent space models for representation learning. In: *International Conference on Machine Learning (ICML)*.

Geng X, Liu H, Lee L, Schuurams D, Levine S and Abbeel P (2022) Multimodal masked autoencoders learn transferable representations. *arXiv preprint arXiv:2205.14204* .

Ghosh D, Walke H, Pertsch K, Black K, Mees O, Dasari S, Hejna J, Xu C, Luo J, Kreiman T, Tan Y, Sadigh D, Finn C and Levine S (2023) Octo: An open-source generalist robot policy.

Goyal R, Kahou SE, Michalski V, Materzynska J, Westphal S, Kim H, Haenel V, Fründ I, Yianilos PN, Mueller-Freitag M, Hoppe F, Thurau C, Bax I and Memisevic R (2017) The "something something video database for learning and evaluating visual common sense. In: *International Conference on Computer Vision (ICCV)*.

Grauman K, Westbury A, Byrne E, Chavis ZQ, Furnari A, Girdhar R, Hamburger J, Jiang H, Liu M, Liu X, Martin M, Nagarajan T, Radosavovic I, Ramakrishnan SK, Ryan F, Sharma J, Wray M, Xu M, Xu EZ, Zhao C, Bansal S, Batra D, Cartillier V, Crane S, Do T, Doulaty M, Erapalli A, Feichtenhofer C, Fragomeni A, Fu Q, Fuegen C, Gebreselasie A, González C, Hillis JM, Huang X, Huang Y, Jia W, Khoo WYH, Kolár J, Kottur S, Kumar A, Landini F, Li C, Li Y, Li Z, Mangalam K, Modhugu R, Munro J, Murrell T, Nishiyasu T, Price W, Puentes PR, Ramazanova M, Sari L, Somasundaram KK, Southerland A, Sugano Y, Tao R, Vo M, Wang Y, Wu X, Yagi T, Zhu Y, Arbeláez P, Crandall DJ, Damen D, Farinella GM, Ghanem B, Ithapu VK, Jawahar CV, Joo H, Kitani K, Li H, Newcombe RA, Oliva A, Park HS, Rehg JM, Sato Y, Shi J, Shou MZ, Torralba A, Torresani L, Yan M and Malik J (2022) Ego4D: Around the world in 3,000 hours of egocentric video. In: *Computer Vision and Pattern Recognition (CVPR)*.

Gupta A, Dollár P and Girshick RB (2019) LVIS: A dataset for large vocabulary instance segmentation. In: *Computer Vision and Pattern Recognition (CVPR)*.

Hafner D, Lillicrap TP, Ba J and Norouzi M (2020) Dream to control: Learning behaviors by latent imagination. In: *International Conference on Learning Representations (ICLR)*.

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

Hauser KK (2012) Recognition, prediction, and planning for assisted teleoperation of freeform tasks. *Autonomous Robots (AURO)* : 241–254.

He K, Chen X, Xie S, Li Y, Dollár P and Girshick RB (2022) Masked autoencoders are scalable vision learners. In: *Computer Vision and Pattern Recognition (CVPR)*.

Hendrycks D and Gimpel K (2016) Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415* .

Hoffman G and Breazeal C (2007) Cost-based anticipatory action selection for human–robot fluency. *IEEE Transactions on Robotics (T-RO)* 23: 952–961.

Hu Y, Wang R, Li L and Gao Y (2023) For pre-trained vision models in motor control, not all policy learning methods are created equal. In: *International Conference on Machine Learning (ICML)*.

Huang W, Wang C, Zhang R, Li Y, Wu J and Fei-Fei L (2023) Voxposer: Composable 3d value maps for robotic manipulation with language models. In: *Conference on Robot Learning (CoRL)*.

Ioffe S and Szegedy C (2015) Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: *International Conference on Machine Learning (ICML)*. pp. 448–456.

James S and Davison AJ (2022) Q-Attention: Enabling efficient learning for vision-based robotic manipulation. *IEEE Robotics and Automation Letters (RA-L)* 7: 1612–1619.

James S, Wada K, Laidlow T and Davison AJ (2022) Coarse-to-fine Q-Attention: Efficient learning for visual robotic manipulation via discretisation. In: *Computer Vision and Pattern Recognition (CVPR)*. pp. 13729–13738.

Jang E, Irpan A, Khansari M, Kappler D, Ebert F, Lynch C, Levine S and Finn C (2021) BC-Z: Zero-shot task generalization with robotic imitation learning. In: *Conference on Robot Learning (CoRL)*.

Javdani S, Admoni H, Pellegrinelli S, Srinivasa SS and Bagnell JA (2018) Shared autonomy via hindsight optimization for teleoperation and teaming. *International Journal of Robotics Research (IJRR)* 37: 717–742.

Jocher G, Stoken A, Borovec J, NanoCode012, ChristopherSTAN, Laughing, tkianai, Hogan A, lorenzomammana, yxNONG, AlexWang1900, Diaconu L, Marc, wanghaoyang0106, ml5ah, Doug, Ingham F, Frederik, Guilhen, Hatovix, Poznanski J, Fang J, Yu L, changyu98, Wang M, Gupta N, Akhtar O, PetrDvoracek and Rai P (2020) *YOLO-v5 repository*.

Jonschkowski R and Brock O (2015) Learning state representations with robotic priors. *Autonomous Robots* 39: 407–428.

Kamath A, Singh M, LeCun Y, Misra I, Synnaeve G and Carion N (2021) MDETR - modulated detection for end-to-end multi-modal understanding. In: *International Conference on Computer Vision (ICCV)*.

Karamcheti S, Orr L, Bolton J, Zhang T, Goel K, Narayan A, Bommasani R, Narayanan D, Hashimoto T, Jurafsky D, Manning CD, Potts C, Ré C and Liang P (2021a) Mistral - a journey towards reproducible language model training.

Karamcheti S, Srivastava M, Liang P and Sadigh D (2021b) LILA: Language-informed latent actions. In: *Conference on Robot Learning (CoRL)*.

Khandelwal A, Weihs L, Mottaghi R and Kembhavi A (2021) Simple but effective: CLIP embeddings for embodied AI. In: *Computer Vision and Pattern Recognition (CVPR)*. pp. 14809–14818.

Kostrikov I, Yarats D and Fergus R (2021) Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. In: *International Conference on Learning Representations (ICLR)*.

Krishna R, Zhu Y, Groth O, Johnson J, Hata K, Kravitz J, Chen S, Kalantidi Y, Li LJ, Shamma DA, Bernstein MS and Li FF (2017) Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision* 123: 32–73.

Laskin M, Lee K, Stooke A, Pinto L, Abbeel P and Srinivas A (2020) Reinforcement learning with augmented data. In: *Advances in Neural Information Processing Systems (NeurIPS)*.

Lee J, Lee Y, Kim J, Kosiorek AR, Choi S and Teh YW (2018) Set transformer: A framework for attention-based permutation-invariant neural networks. In: *International Conference on Machine Learning (ICML)*.

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

Li Y, Mao H, Girshick RB and He K (2022) Exploring plain vision transformer backbones for object detection. In: *European Conference on Computer Vision (ECCV)*.

Lin TY, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dollár P and Zitnick CL (2014) Microsoft COCO: Common objects in context. In: *European Conference on Computer Vision (ECCV)*. pp. 740–755.

Liu H, Lee L, Lee K and Abbeel P (2022) Instructrl: Simple yet effective instruction-following agents with multimodal transformer. *arXiv preprint arXiv:2210.13431* .

Liu H, Nasiriany S, Zhang L, Bao Z and Zhu Y (2023a) Robot learning on the job: Human-in-the-loop autonomy and learning during deployment. In: *Robotics: Science and Systems (RSS)*.

Liu S, Zeng Z, Ren T, Li F, Zhang H, Yang J, yue Li C, Yang J, Su H, Zhu JJ and Zhang L (2023b) Grounding DINO: Marrying DINO with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499* .

Lu J, Batra D, Parikh D and Lee S (2019) ViLBERT: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In: *Advances in Neural Information Processing Systems (NeurIPS)*.

Lu J, Clark C, Zellers R, Mottaghi R and Kembhavi A (2023) Unified-io: A unified model for vision, language, and multi-modal tasks. In: *International Conference on Learning Representations (ICLR)*.

Lynch C and Sermanet P (2020) Grounding language in play. *arXiv preprint arXiv:2005.07648* .

Ma YJ, Liang W, Som V, Kumar V, Zhang A, Bastani O and Jayaraman D (2023) LIV: Language-image representations and rewards for robotic control. In: *International Conference on Machine Learning (ICML)*.

Ma YJ, Sodhani S, Jayaraman D, Bastani O, Kumar V and Zhang A (2022) VIP: Towards universal visual reward and representation via value-implicit pre-training. In: *International Conference on Learning Representations (ICLR)*.

Mahler J, Liang J, Niyaz S, Laskey M, Doan R, Liu X, Ojea JA and Goldberg K (2017) Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics. In: *Robotics: Science and Systems (RSS)*.

Majumdar A, Yadav K, Arnaud S, Ma YJ, Chen C, Silwal S, Jain A, Berges VP, Abbeel P, Malik J, Batra D, Lin Y, Maksymets O,

Rajeswaran A and Meier F (2023) Where are we in the search for an artificial visual cortex for embodied intelligence? *arXiv preprint arXiv:2303.18240* .

Mandlekar A, Xu D, Wong J, Nasiriany S, Wang C, Kulkarni R, Fei-Fei L, Savarese S, Zhu Y and Martín-Martín R (2021) What matters in learning from offline human demonstrations for robot manipulation. In: *Conference on Robot Learning (CoRL)*.

Mees O, Borja-Diaz J and Burgard W (2023) Grounding language with visual affordances over unstructured data. In: *International Conference on Robotics and Automation (ICRA)*.

Miech A, Zhukov D, Alayrac JB, Tapaswi M, Laptev I and Sivic J (2019) HowTo100M: Learning a text-video embedding by watching hundred million narrated video clips. In: *International Conference on Computer Vision (ICCV)*. pp. 2630–2640.

Minderer M, Gritsenko AA and Houlsby N (2023) Scaling open-vocabulary object detection. In: *Advances in Neural Information Processing Systems (NeurIPS)*.

Minderer M, Gritsenko AA, Stone A, Neumann M, Weissenborn D, Dosovitskiy A, Mahendran A, Arnab A, Dehghani M, Shen Z, Wang X, Zhai X, Kipf T and Houlsby N (2022) Simple open-vocabulary object detection with vision transformers. In: *European Conference on Computer Vision (ECCV)*.

Misra DK, Langford J and Artzi Y (2017) Mapping instructions and visual observations to actions with reinforcement learning. In: *Empirical Methods in Natural Language Processing (EMNLP)*.

Nair S, Mitchell E, Chen K, Ichter B, Savarese S and Finn C (2021) Learning language-conditioned robot behavior from offline data and crowd-sourced annotation. In: *Conference on Robot Learning (CoRL)*.

Nair S, Rajeswaran A, Kumar V, Finn C and Gupta A (2022) R3m: A universal visual representation for robot manipulation. *arXiv preprint arXiv:2203.12601* .

Narang S, Chung HW, Tay Y, Fedus W, Févry T, Matena M, Malkan K, Fiedel N, Shazeer NM, Lan Z, Zhou Y, Li W, Ding N, Marcus J, Roberts A and Raffel C (2021) Do transformer modifications transfer across implementations and applications? In: *Empirical Methods in Natural Language Processing (EMNLP)*.

Nasiriany S, Gao T, Mandlekar A and Zhu Y (2022) Learning and retrieval from prior data for skill-based imitation learning. In: *Conference on Robot Learning (CoRL)*.

OpenAI (2022) Chatgpt: Optimizing language models for dialogue.

Oquab M, Darcet T, Moutakanni T, Vo HQ, Szafraniec M, Khalidov V, Fernandez P, Haziza D, Massa F, El-Nouby A, Assran M, Ballas N, Galuba W, Howes R, Huang PYB, Li SW, Misra I, Rabbat MG, Sharma V, Synnaeve G, Xu H, Jégou H, Mairal J, Labatut P, Joulin A and Bojanowski P (2023) DINOv2: Learning robust visual features without supervision. *Transactions of Machine Learning Research (TMLR)* .

Padalkar A, Pooley A, Jain A, Bewley A, Herzog A, Irpan A, Khazatsky A, Rai A, Singh A, Brohan A, Raffin A, Wahid A, Burgess-Limerick B, Kim B, Schölkopf B, Ichter B, Lu C, Xu C, Finn C, Xu C, Chi C, Huang C, Chan C, Pan C, Fu C, Devin C, Driess D, Pathak D, Shah D, Büchler D, Kalashnikov D, Sadigh D, Johns E, Ceola F, Xia F, Stulp F, Zhou G, Sukhatme GS, Salhotra G, Yan G, Schiavi G, Su H, Fang H, Shi H, Amor HB, Christensen HI, Furuta H, Walke H, Fang H, Mordatch I, Radosavovic I, Leal I, Liang J, Kim J, Schneider J, Hsu J, Bohg J, Bingham J, Wu J, Wu J, Luo J, Gu J, Tan J, Oh J, Malik J, Tompson J, Yang J, Lim JJ, Silvério J, Han J, Rao K, Pertsch K, Hausman K, Go K, Gopalakrishnan K, Goldberg K, Byrne K, Oslund K, Kawaharazuka K, Zhang K, Majd K, Rana K, Srinivasan KP, Chen LY, Pinto L, Tan L, Ott L, Lee L, Tomizuka M, Du M, Ahn M, Zhang M, Ding M, Srirama MK, Sharma M, Kim MJ, Kanazawa N, Hansen N, Heess NMO, Joshi NJ, Suenderhauf N, Palo ND, Shafiullah NMM, Mees O, Kroemer O, Sanketi PR, Wohlhart P, Xu P, Sermanet P, Sundaresan P, Vuong QH, Rafailov R, Tian R, Doshi R, Mendonca R, Shah R, Hoque R, Julian RC, Bustamante S, Kirmani S, Levine S, Moore S, Bahl S, Dass S, Song S, Xu S, Haldar S, Adebola SO, Guist S, Nasiriany S, Schaal S, Welker S, Tian S, Dasari S, Belkhale S, Osa T, Harada T, Matsushima T, Xiao T, Yu T, Ding T, Davchev T, Zhao T, Armstrong T, Darrell T, Jain V, Vanhoucke V, Zhan W, Zhou W, Burgard W, Chen X, Wang X, Zhu X, Li X, Lu Y, Chebotar Y, Zhou Y, Zhu Y, Xu Y, Wang Y, Bisk Y, Cho Y, Lee Y, Cui Y, hua Wu Y, Tang Y, Zhu Y, Li Y, Iwasawa Y, Matsuo Y, Xu Z and Cui ZJ (2023) Open X-Embodiment: Robotic learning datasets and RT-X models. *arXiv preprint arXiv:2310.08864* .

Pari J, Shafiullah NMM, Arunachalam SP and Pinto L (2022) The surprising effectiveness of representation learning for visual imitation. In: *Robotics: Science and Systems (RSS)*.

Parisi S, Rajeswaran A, Purushwalkam S and Gupta AK (2022) The unsurprising effectiveness of pre-trained vision models for control. *arXiv preprint arXiv:2203.03580* .

Radford A, Kim JW, Hallacy C, Ramesh A, Goh G, Agarwal S, Sastry G, Askell A, Mishkin P, Clark J, Krueger G and Sutskever I (2021) Learning transferable visual models from natural language supervision. In: *International Conference on Machine Learning (ICML)*, volume 139. pp. 8748–8763.

Radosavovic I, Xiao T, James S, Abbeel P, Malik J and Darrell T (2022) Real-world robot learning with masked visual pre-training. In: *Conference on Robot Learning (CoRL)*.

Raffel C, Shazeer N, Roberts A, Lee K, Narang S, Matena M, Zhou Y, Li W and Liu PJ (2019) Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683* .

Raghu M, Unterthiner T, Kornblith S, Zhang C and Dosovitskiy A (2021) Do vision transformers see like convolutional neural networks? In: *Advances in Neural Information Processing Systems (NeurIPS)*.

Ramanathan V, Kalia A, Petrovic V, Wen Y, Zheng B, Guo B, Wang R, Marquez A, Kovvuri R, Kadian A, Mousavi A, Song YZ, Dubey A and Mahajan DK (2023) PACO: Parts and attributes of common objects. In: *Computer Vision and Pattern Recognition (CVPR)*.

Reed S, Zolna K, Parisotto E, Colmenarejo SG, Novikov A, Barth-Maron G, Gimenez M, Sulsky Y, Kay J, Springenberg JT, Eccles T, Bruce J, Razavi A, Edwards AD, Heess NMO, Chen Y, Hadsell R, Vinyals O, Bordbar M and de Freitas N (2022) A generalist agent. *Transactions of Machine Learning Research (TMLR)* .

Reid M, Yamada Y and Gu SS (2022) Can Wikipedia help offline reinforcement learning? *arXiv preprint arXiv:2201.12122* .

Rezatofighi SH, Tsoi N, Gwak J, Sadeghian A, Reid ID and Savarese S (2019) Generalized intersection over union: A metric and a loss for bounding box regression. In: *Computer Vision and Pattern Recognition (CVPR)*.

Rosete-Beas E, Mees O, Kalweit G, Boedecker J and Burgard W (2022) Latent plans for task agnostic offline reinforcement learning. In: *Conference on Robot Learning (CoRL)*.

Sanh V, Debut L, Chaumond J and Wolf T (2019) DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108* .

Saxena A, Driemeyer J and Ng A (2008) Robotic grasping of novel objects using vision. *International Journal of Robotics Research (IJRR)* 27: 157–173.

Saxena S, Sharma M and Kroemer O (2023) Multi-resolution sensing for real-time control with vision-language models. In: *Conference on Robot Learning (CoRL)*.

Sermanet P, Lynch C, Chebotar Y, Hsu J, Jang E, Schaal S and Levine S (2018) Time-contrastive networks: Self-supervised learning from video. In: *International Conference on Robotics and Automation (ICRA)*. pp. 1134–1141.

Shah D, Sridhar AK, Dashora N, Stachowicz K, Black K, Hirose N and Levine S (2023) ViNT: A foundation model for visual navigation. In: *Conference on Robot Learning (CoRL)*.

Shah R and Kumar V (2021) Rrl: Resnet as representation for reinforcement learning. In: *International Conference on Machine Learning (ICML)*.

Shan D, Geng J, Shu M and Fouhey DF (2020) Understanding human hands in contact at internet scale. In: *Computer Vision and Pattern Recognition (CVPR)*. pp. 9866–9875.

Shao L, Migimatsu T, Zhang Q, Yang K and Bohg J (2020) Concept2robot: Learning manipulation concepts from instructions and human demonstrations. In: *Robotics: Science and Systems (RSS)*.

Shao S, Li Z, Zhang T, Peng C, Yu G, Zhang X, Li J and Sun J (2019) Objects365: A large-scale, high-quality dataset for object detection. In: *International Conference on Computer Vision (ICCV)*.

Shazeer NM (2020) GLU variants improve transformer. *arXiv preprint arXiv:2002.05202* .

Shridhar M, Manuelli L and Fox D (2021) Cliport: What and where pathways for robotic manipulation. In: *Conference on Robot Learning (CoRL)*.

Shridhar M, Manuelli L and Fox D (2022) Perceiver-actor: A multi-task transformer for robotic manipulation. In: *Conference on Robot Learning (CoRL)*.

Singh A, Hu R, Goswami V, Couairon G, Galuba W, Rohrbach M and Kiela D (2022) FLAVA: A foundational language and vision alignment model. In: *Computer Vision and Pattern Recognition (CVPR)*. pp. 15617–15629.

Smith L, Dhawan N, Zhang M, Abbeel P and Levine S (2020) Avid: Learning multi-stage tasks via pixel-level translation of human videos. In: *Robotics: Science and Systems (RSS)*.

Srinivas A, Laskin M and Abbeel P (2020) CURL: Contrastive unsupervised representations for reinforcement learning. In: *International Conference on Machine Learning (ICML)*.

Stepputtis S, Campbell J, Phielipp M, Lee S, Baral C and Amor HB (2020) Language-conditioned imitation learning for robot manipulation tasks. In: *Advances in Neural Information Processing Systems (NeurIPS)*.

Strudel R, Pinel RG, Laptev I and Schmid C (2021) Segmenter: Transformer for semantic segmentation. In: *International Conference on Computer Vision (ICCV)*. pp. 7242–7252.

Tellex S, Kollar T, Dickerson S, Walter MR, Banerjee AG, Teller SJ and Roy N (2011) Understanding natural language commands for robotic navigation and mobile manipulation. In: *Association for the Advancement of Artificial Intelligence (AAAI)*.

Tong Z, Song Y, Wang J and Wang L (2022) VideoMAE: Masked autoencoders are data-efficient learners for self-supervised video pre-training. In: *Advances in Neural Information Processing Systems (NeurIPS)*.

Touvron H, Cord M, Sablayrolles A, Synnaeve G and Jégou H (2021) Going deeper with image transformers. In: *International Conference on Computer Vision (ICCV)*. pp. 32–42.

Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L and Polosukhin I (2017) Attention is all you need. *arXiv preprint arXiv:1706.03762* .

Vaswani A, Zhao Y, Fossum V and Chiang D (2013) Decoding with large-scale neural language models improves translation. In: *Empirical Methods in Natural Language Processing (EMNLP)*. pp. 1387–1392.

Walke H, Black K, Lee A, Kim MJ, Du M, Zheng C, Zhao T, Hansen-Estruch P, Vuong Q, He A, Myers V, Fang K, Finn C and Levine S (2023) Bridgedata v2: A dataset for robot learning at scale. In: *Conference on Robot Learning (CoRL)*.

Wang CY, Bochkovskiy A and Liao HYM (2022) YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In: *Computer Vision and Pattern Recognition (CVPR)*.

Wang KJ, Liu YH, Su HT, Wang JW, Wang YS, Hsu WH and Chen WC (2021) OCID-Ref: A 3d robotic dataset with embodied language for clutter scene grounding. In: *Association for Computational Linguistics (ACL)*.

Weiss LE, Sanderson AC and Neuman CP (1987) Dynamic sensor-based control of robots with visual feedback. *IEEE Robotics and Automation Letters (RA-L)* 3: 404–417.

Wightman R (2019) Pytorch image models. `https://github.com/rwightman/pytorch-image-models`.

Wu J, Antonova R, Kan A, Lepert M, Zeng A, Song S, Bohg J, Rusinkiewicz S and Funkhouser TA (2023) Tidybot: Personalized robot assistance with large language models. In: *International Conference on Intelligent Robots and Systems (IROS)*.

Xiao T, Radosavovic I, Darrell T and Malik J (2022) Masked visual pre-training for motor control. *arXiv preprint arXiv:2203.06173* .

Yu J, Wang Z, Vasudevan V, Yeung L, Seyedhosseini M and Wu Y (2022) Coca: Contrastive captioners are image-text foundation models. *Transactions of Machine Learning Research (TMLR)* .

Yu L, Poirson P, Yang S, Berg AC and Berg TL (2016) Modeling context in referring expressions. In: *European Conference on Computer Vision (ECCV)*.

Zeng A, Song S, Yu KT, Donlon E, Hogan FR, Bauzá M, Ma D, Taylor O, Liu M, Romo E, Fazeli N, Alet F, Dafle NC, Holladay R, Morona I, Nair PQ, Green D, Taylor I, Liu W, Funkhouser TA and Rodriguez A (2017) Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching. *International Journal of Robotics Research (IJRR)* 41: 690–705.

Zeng A, Wong AS, Welker S, Choromanski K, Tombari F, Purohit A, Ryoo MS, Sindhwani V, Lee J, Vanhoucke V and Florence PR (2023) Socratic models: Composing zero-shot multimodal reasoning with language. In: *International Conference on Learning Representations (ICLR)*.

Zhai X, Kolesnikov A, Houlsby N and Beyer L (2022) Scaling vision transformers. In: *Computer Vision and Pattern Recognition (CVPR)*. pp. 1204–1213.

Zhang A, McAllister R, Calandra R, Gal Y and Levine S (2021) Learning invariant representations for reinforcement learning without reconstruction. In: *International Conference on Learning Representations (ICLR)*.

Zhang B and Sennrich R (2019) Root mean square layer normalization. In: *Advances in Neural Information Processing Systems (NeurIPS)*.

Zheng S, Lu J, Zhao H, Zhu X, Luo Z, Wang Y, Fu Y, Feng J, Xiang T, Torr PHS and Zhang L (2021) Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In: *Computer Vision and Pattern Recognition (CVPR)*.

Zhou T, Tucker R, Flynn J, Fyffe G and Snavely N (2018) Stereo magnification: Learning view synthesis using multiplane images. In: *Special Interest Group on Computer Graphics and Interactive Techniques (SIGGRAPH)*.

Zhu C, Xiao F, Alvarado A, Babaei Y, Hu J, El-Mohri H, Culatana SC, Sumbaly R and Yan Z (2023) Egoobjects: A large-scale egocentric dataset for fine-grained object understanding. *arXiv preprint arXiv:2309.08816* .

Overview

In the appendices below, we provide additional details around the implementation, pretraining, and adaptation procedures described in the main text, in addition to delving deeper into various discussions. Finally, we add additional results and visualizations that further complement the findings from the main text.

We provide open-source code for loading and using pretraining models, hosted links for our preprocessing splits (including the actual batches seen during training), and a separate, standalone open-source code repository for our evaluation suite. Our hope is that the evaluation suite especially is general and easy to use for downstream work on evaluating learned representations.

The full manifest of resources are as follows:

- Open-Source Models / Training Repository: `https://github.com/siddk/voltron-robotics`
- Open-Source Evaluation Suite (API for automated evaluation): `https://github.com/siddk/voltron-evaluation`
- Project Page (videos & additional links): `https://sites.google.com/view/voltron-robotics`

An overview of each appendix can be found below. We further indicate which parts of the appendices are best viewed here in the text or on the project page; for videos and visualizations, we highly recommend navigating to the latter.

A Voltron Implementation & Artifacts

We provide complete implementation details for the various Voltron models, from the small modifications to the Transformer block for added pretraining stability, to the added structural components for embedding multimodal (vision and language) inputs. All of these details are made explicit in our code release, linked on our project page.

```python
class StandardBlock:
  def __init__(self, embed_dim: int, n_heads: int, mlp_dim: int) -> None:
    """
    Standard ViT Transformer Block with LayerNorm and GELU() activation.

    A Transformer Block consists of two normalization layers, applied prior to
    the multiheaded attention module (standard) and position-wise feed-forward
    MLP (standard) respectively.
    """
    self.pre_norm_attn = nn.LayerNorm(embed_dim, eps=1e-6)
    self.attn = nn.MultiHeadAttention(embed_dim, n_heads=n_heads)

    self.pre_norm_mlp = nn.LayerNorm(embed_dim, eps=1e-6)
    self.mlp = nn.Sequential(
      nn.Linear(embed_dim, mlp_dim),
      nn.GELU(),
      nn.Linear(mlp_dim, embed_dim),
    )

    # A Standard Block has no other components...

  def forward(self, x: T[bsz, seq, embed_dim]) -> T[bsz, seq, embed_dim]:
    x = x + self.attn(self.pre_norm_attn(x))
    x = x + self.mlp(self.pre_norm_mlp(x))
    return x
```

```python
class VoltronBlock:
  def __init__(self, embed_dim: int, n_heads: int, mlp_dim: int) -> None:
    """
    Voltron Transformer Block with RMSNorm, SwishGLU() activation, and
    LayerScale.

    A Transformer Block consists of two normalization layers, applied prior to
    the multiheaded attention module (standard) and position-wise feed-forward
    MLP (standard) respectively; adds residual LayerScale as well!
    """
    self.pre_norm_attn = RMSNorm(embed_dim) # RMSNorm is simpler/more stable
    self.attn = nn.MultiHeadAttention(embed_dim, n_heads=n_heads)

    self.pre_norm_mlp = RMSNorm(embed_dim)
    self.mlp = nn.Sequential(
      SwishGLU(embed_dim, mlp_dim), # SwishGLU includes a learned projection
      nn.Linear(mlp_dim, embed_dim),
    )

    self.scale_attn = LayerScale(embed_dim)
    self.scale_mlp = LayerScale(embed_dim)

  def forward(self, x: T[bsz, seq, embed_dim]) -> T[bsz, seq, embed_dim]:
    x = x + self.scale_attn(self.attn(self.pre_norm_attn(x)))
    x = x + self.scale_mlp(self.mlp(self.pre_norm_mlp(x)))
    return x
```

```python
class SwishGLU:
  def __init__(self, in_dim: int, out_dim: int) -> None:
    """
    SwishGLU as defined in PaLM (Google).
      => Ref: "GLU Variants Improve Transformers" (Shazeer)
    """
    self.project = nn.Linear(in_dim, out_dim)
    self.gate =  nn.Linear(in_dim, out_dim)

  def forward(self: x: T[..., in_dim]) -> T[..., out_dim]:
    projected = self.project(x)
    gate = self.gate(x)
    return projected * nn.SiLU(gate)
```

```python
class RMSNorm:
  def __init__(self, dim: int) -> None:
    self.g, self.dim = nn.Parameter(T.ones(dim)), dim

  def forward(self: x: T[..., dim]) -> T[..., dim]:
    return self.g * (x / (norm(x) / sqrt(self.dim)))

class LayerScale:
  def __init__(self, dim: int, init: float = 0.1) -> None:
    self.gamma = nn.Parameter(init * T.ones(dim))

  def forward(self: x: T[..., dim]) -> T[..., dim]:
    return self.gamma * x
```

Figure 12. Standard vs. Voltron Transformer Implementation. The Voltron Transformer Block is near-identical to the "standard" Transformer block used in prior work in Vision Transformers, with exceptions marked in orange. Notably, we switch `LayerNorm` for `RMSNorm`, a standard MLP with a `GELU` activation (Hendrycks and Gimpel 2016) with a `SwishGLU` activation, and adopt `LayerScale` for each residual connection; these components are defined explicitly below the block definitions. In ablating these architecture modifications, we find *no impact on downstream performance, but increased pretraining stability*.

A.1 Voltron Transformer Implementation

As mentioned in §4, we perform a series of modifications to the typical Transformer block used in prior work in the Vision Transformer and Masked Autoencoding literature to help with pretraining stability; these changes are motivated by recent work from the NLP community on training stable and performant Transformer models (Narang et al. 2021; Karamcheti et al. 2021a; Chowdhery et al. 2022).

We show the side-by-side comparison of the "standard" Transformer block implementation vs. the Voltron Transformer block in Fig. 12. The changes are three-fold:

- Using Root Mean-Square Normalization (Zhang and Sennrich 2019) over the default LayerNorm; not only does RMSNorm have fewer parameters, but it has been shown to increase stability and performance (Narang et al. 2021).
- Using the SwishGLU activation (Shazeer 2020; Chowdhery et al. 2022) over the default GELU (Hendrycks and Gimpel 2016).
- Using LayerScale (Touvron et al. 2021) for scaling down the magnitude of residual connections; prior work has found this to have a powerful stabilizing effect during pretraining (Karamcheti et al. 2021a).

We also provide pseudocode for implementing the various modifications in Fig. 12 (bottom); these modifications are all simple and transferable across Transformer implementations. Furthermore, we ablate the effects of these modifications on performance; we find *that these modifications do not change downstream performance, but significantly increase pretraining stability*, following our initial motivation.

A.2 Jointly Processing Vision & Language

To incorporate language into the typical masked autoencoding pipeline, we add a series of small structural changes to handle 1) multi-modality, 2) sharing a Transformer decoder for both visual reconstruction and language generation, and 3) handling position encoding for both visual patch embeddings and textual tokens.

Figure 13. Voltron Pretraining Learning Curves (Reconstruction Error). We visualize the reconstruction error over pretraining epoch for each of the Voltron models. Note that each model learns differently, converging to different reconstruction errors: both the language-conditioned models ($\alpha = 0$) converge to low reconstruction error, with $\mathcal{V}$**-Dual** showing that encoding and learning over multi-frame contexts allowing for a better fit. The language generative model $\mathcal{V}$**-Gen** ($\alpha = 0.5$) converges to a relatively higher reconstruction error, showing the tension between balancing two disparate objectives.

Multimodal Encoder. We make the following adjustments to enable a Transformer encoder to embed multiple modalities. First, we project both our learned "patch embeddings" (obtained as in a standard ViT, by learning a linear transformation of our flattened RGB patches of size $p \times p \times 3$) and our pretrained language embeddings to the same space $\mathbb{R}^d$, where d is the Transformer dimensionality (e.g., $d = 384$ for a ViT-Small). While we learn our patch embedding end-to-end, we initialize our language embeddings from a pretrained (and frozen) DistilBERT model (Sanh et al. 2019); this is following R3M (Nair et al. 2022). We pad each language annotation c in our dataset to a maximum length $L = 20$ tokens, additionally storing a binary length mask to ensure that each Transformer block does not attend to padding.

Once projected into the Transformer's embedding space, we add learned modality embeddings (e.g., an embedding for `<IMG>` and `<LANG>`) to each of the respective inputs; we find that this better allows the Transformer to reason over different modalities. We initialize these learnable embeddings via a truncated normal distribution, with scale $\sigma = 0.02$, following how other special embeddings are initialized in the MAE and Vision Transformer literature (He et al. 2022).

The final step is for handling multi-frame contexts; we learn a set of frame index embeddings (e.g., for `FRAME-1`, `FRAME-2`, etc.) and add these to the corresponding patch embeddings – i.e. we add the `FRAME-i` embedding to all patch embeddings from the first frame and so on. This further allows us to distinguish individual frame patches from one another.

At this point, we concatenate the full sequence of flattened visual patch embeddings and language token embeddings, and feed them through the stack of Transformer blocks that form the multimodal encoder. This output is fed to the decoder, in the same fashion as a traditional masked autoencoder.

Shared Transformer for Reconstruction & Generation. As mentioned in §4, we make one crucial change to the standard Transformer decoder in a masked autoencoder to additionally allow for language generation: namely adding a *prefix mask over the language inputs* (Raffel et al. 2019). The goal of this mask (as stated in the main text) is to prevent information leakage when decoding; this mask selectively zeroes out dependencies in the multiheaded attention during training such that when generating language given a visual context, each language embedding at a given timestep t can only attend to prior generated language at timesteps $< t$, as well as the entire visual context. This masking operates in the same way as the original decoder masking described in Vaswani et al. (2013); the attention scores for all "invalid" inputs ($> t$) are set to 0, restricting the model from incorporating future predictions as it processes the sequence.

Apart from this, the only other change we make to the MAE decoder is learning a separate set of modality embeddings (as described in the prior section) – i.e. embeddings for `<IMG-DECODER>` and `<LANG-DECODER>`; the reason for this is that the Decoder sees a series of `<MASK>` embeddings representing the "unseen" visible context to reconstruct, as well as the new language context to generate (recall that because of the α gating, the language generator *never* sees language embeddings from the encoder). We add these to the corresponding embeddings fed to the decoder, then resume the standard MAE decoding pipeline and the language generation pipeline (autoregressively generating the original annotation).

Position Encoding. We follow standard pratice in the masked autoencoding literature (and the same practice used by MVP), as position encode each of the patch embeddings subject to a fixed (deterministic) 2D sinusoidal embedding that reflects both vertical and horizontal positioning of each patch within a grid – this is taken directly from the original MAE codebase. To encode text, we use a similar strategy, using a 1D sinusoidal embedding added to each token embedding in a sequence.

A.3 Pretraining Curves

To further contextualize our results and enrich the discussion earlier in the paper (and further on in the appendices), we include the pretraining loss curves for each of the three Voltron models we train in this work – $\mathcal{V}$**-Cond**, $\mathcal{V}$**-Dual**, and $\mathcal{V}$**-Gen**. The reconstruction error curves for the three models can be found in Fig. 13. In general, we find that the "trade-off" between language-conditioned reconstruction and visually-grounded language generation is made concrete in the pretraining loss – both purely language-conditioned models ($\mathcal{V}$**-Cond**, $\mathcal{V}$**-Dual** with $\alpha = 0$) converge to fairly low reconstruction error; however, $\mathcal{V}$**-Gen** (with $\alpha = 0.5$) converges to a much higher reconstruction error – due to the tension between optimizing for both reconstruction and language generation. We additionally note that adding even simple, dual-frame contexts enables lower reconstruction error – even with the ViT-Small models, on the Sth-Sth dataset.

A.4 Adapting Representations for Evaluation

The description of the adaptation pipeline described in §5 outlines all major details for the adaptation experiments for each evaluation domain; the role of this section is to clarify any potentially ambiguous details, and further motivate some of the choices we make in implementing each evaluation. In general, we keep the adaptation architecture and optimization parameters as simple as possible. For all applications we use a default learning rate of 1e-3, and weight decay of 0.01.

Grasp Affordance Prediction. We implement the adaptation head for the grasp affordance prediction task following recent work in learning segmentation heads on top of vision transformer features, specifically following the procedure outlined in Segmentation Transformers via Progressive Upsampling (SETR-PUP) (Zheng et al. 2021). A PUP block is straightforward – we first extract all patch embeddings from the output of our Vision Transformer encoder, using a shallow MAP block with the same number of seed vectors as patches output by the encoder. We then reshape the extracted features into a *grid*, then stack a series of 4 upsampling blocks (channel depths of $[128, 64, 32, 16]$, ReLU activation) that consist of a 2D convolution followed by a bilinear upsampling, until we recover a grid of the same size of the original image. We finally apply a spatial softmax, predicting distributions over each of the possible labels ("graspable," "non-graspable," "background"), and compute our loss per-pixel. We optimize with a batch size of 64, for 50 epochs in total. Given the small size of the dataset, we find that there is a great deal of variance across random initializations; we report results by running 5-fold cross-validation, taking the model with the best performance across validation folds to compute final test statistics.

Referring Expression Grounding. We use a simple adaptation head for referring expression grounding that extracts a single dense representation from our learned encoder via a shallow MAP block with a single seed vector (the default extractor for obtaining a vector representation of a visual input). For representations that are not language-conditioned, we concatenate this vector with the language embedding under the appropriate model – e.g., the CLIP text embedding for *CLIP (ViT-Base)* – or the DistilBERT language embedding for pure visual models (e.g., MVP). We then feed this context through a 4-layer MLP (hidden dimensions of $[512, 128, 128, 64]$, GELU activation) that directly predicts bounding box coordinates as $(x, y, \text{width}, \text{height})$. We use a Huber loss to compute error. We optimize with a batch size of 512, for 10 epochs in total, using the provided validation set for model selection.

Single-Task Visuomotor Control. We first extract a dense representation using a shallow MAP block (as described above), then follow the exact procedure for evaluating both Franka Kitchen and Adroit policy learning as described in the R3M work (Nair et al. 2022). Namely, we concatenate the visual representation with the robot's proprioceptive state, followed by a BatchNorm layer (Ioffe and Szegedy 2015). These are then fed to a 2-layer MLP ($d = 256$) that directly predicts action targets for computing mean-squared error against the ground-truth actions. Following R3M, we run 20,000 gradient steps with a batch size of 32, evaluating the models online every 5000 steps on a heldout set of 50 environments (fixed seed) – we report success rate subject to the best performing model from the online evaluation. We run three seeds for each combination of viewpoint, number of demonstrations, and task.

Real-World Language-Conditioned Imitation. The full set of language instructions generated by ChatGPT can be found on our project page. For adaptation, we first extract a representation as with the referring expression evaluation by using a shallow MAP block, and concatenating the corresponding language embedding as appropriate. We concatenate this fused vector with the robot's proprioceptive state, and pass the corresponding embedding to a BatchNorm layer. Then, following recent work on real-world imitation learning (Mandlekar et al. 2021), we only train a shallow 2-layer MLP with ($d = 64$) to predict action targets for computing mean-squared error against the ground-truth waypoint actions. We optimize with a batch size of 256, and train for 10 epochs. As policy evaluation in the real-world is expensive – especially for the five approaches we evalaute – we uniformly choose the last epoch checkpoint to perform evaluation rollouts.

Qualitative: Zero-Shot Intent Scoring. This is a zero-shot evaluation with no adaptation data, only applicable to the representation learning models capable of "scoring" joint vision-language contexts: $\mathcal{V}$**-Gen**, *CLIP (ViT-Base)*, and *R3M (Ego4D)*. We download videos from the WHiRL dataset off of the WHiRL website: `https://human2robot.github.io/`.

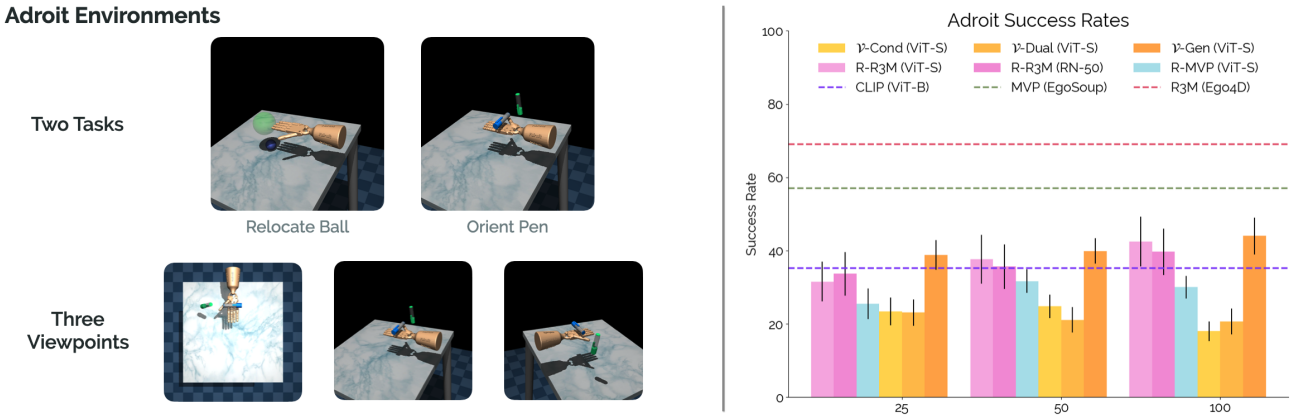

Figure 14. Adroit – Single-Task Visuomotor Control Results. Visualization of the high-dimensional Adroit environments, comprised of two dexterous manipulation tasks, with three camera viewpoints **[Left]**. Results (success rate for each of n demonstrations with $n \in [25, 50, 100]$) for Voltron and baselines (over 3 seeds) **[Right]**. Note the flipped trends relative to the Franka Kitchen results – notably, the more "high-level" representations (from CLIP, R3M, or $\mathcal{V}$**-Gen**) tend to do better on this task; yet, $\mathcal{V}$**-Gen** is still outperforming R-R3M and CLIP, showing the benefit of language-driven flexible learning.

B Additional Results & Visualizations

We present additional results and visualizations to further support our claims from the main text. We provide additional discussion of 1) additional single task control results on the Adroit dexterous manipulation environments, 2) qualitative trajectory rollouts from the $\mathcal{V}$**-Gen** language-conditioned imitation policy, and 3) additional qualitative intent scoring results.

B.1 Results: Adroit Visuomotor Control

To supplement our single-task visuomotor control results, we run out evaluations on the Adroit dexterous manipulation tasks from the R3M paper (Nair et al. 2022). The two tasks we evaluate on, depicted in Fig. 14 (left) consist of controlling a high degree-of-freedom robotic hand (24-DoF) for the task of 1) relocating a ball on the table to a specified target position, and 2) reorienting a pen within the hand to reach a target orientation. Given the innate difficulty of controlling a high-dimensional dexterous robotic hand over a 9-DoF fixed arm manipulator, these tasks are evaluated with $n \in [25, 50, 100]$ demonstrations instead of $n \in [5, 10, 25]$ as with the Franka Kitchen evaluation. In general, learning policies in this environment is *difficult*, especially from limited data.

Looking to the results we see that on this environment, $\mathcal{V}$**-Gen** and R-R3M models tend to be the most performant, in contrast with the Franka Kitchen results which favored $\mathcal{V}$**-Cond** and $\mathcal{V}$**-Dual** (the reconstruction-leaning models). Interestingly, this flipped trend seems to suggest that even within single-task control, different tasks and environments seems to prefer different visual features to perform well – in this case, the more high-level features under models such as R-R3M and $\mathcal{V}$**-Gen** seem to be preferred. In a way, this makes sense; unlike with Franka Kitchen, the actual background objects and interactions thereof – turning knobs, opening microwaves, or sliding doors with clearly marked handles – seem more sensitive to low-level features (where on the microwave is the handle, which knob of the various possible needs to be turned). In Adroit however, these tasks are on clean backgrounds, with individual objects; the high-level behaviors instead that are more important (e.g., "is the ball getting closer to the target location?"). It would be an interesting direction for future work to further profile other "common" visuomotor control tasks along this axis, to get a better understanding of what visual representations must capture to be generally useful (predictive of performance on downstream real-world control problems).

B.2 Qualitative: Real-Robot Language-Conditioned Policy Rollouts

While the experimental results in §5 capture the quantitative success rates of various methods for language-conditioned imitation, they do not paint a picture of *how* these policies behave. In Fig. 15 we show three different rollouts for the best-performing $\mathcal{V}$**-Gen** model: a task success (in-distribution), a task failure (in-distribution), and an example rollout from the visual distractor split. With the waypoint-based action space described in §5, we generally see smooth motions; however, the failure mode of these policies are "oscillations" (Fig. 15; middle) where the policy collapses to predicting the same two waypoints repeatedly. *Full videos of rollouts from each representation learning approach* are all on our project page.

B.3 Qualitative: Additional Intent Scoring Visualizations

Fig. 16 presents additional intent scoring qualitative visualizations for two other tasks from the WHiRL dataset (Bahl et al. 2022) – specifically "lifting the lid off a pot" and "stacking cups." In both scenarios, we see similar behavior to the results from §V of the main text: $\mathcal{V}$**-Gen** shows a propensity for not only tracking the key progress points in the videos for *both human and robot* agents, but also providing a dense and smooth measure of intermediate progress. Both *CLIP (ViT-Base)* and *R3M (Ego4D)* unfortunately predict high-variance scores, seemingly random across the video.

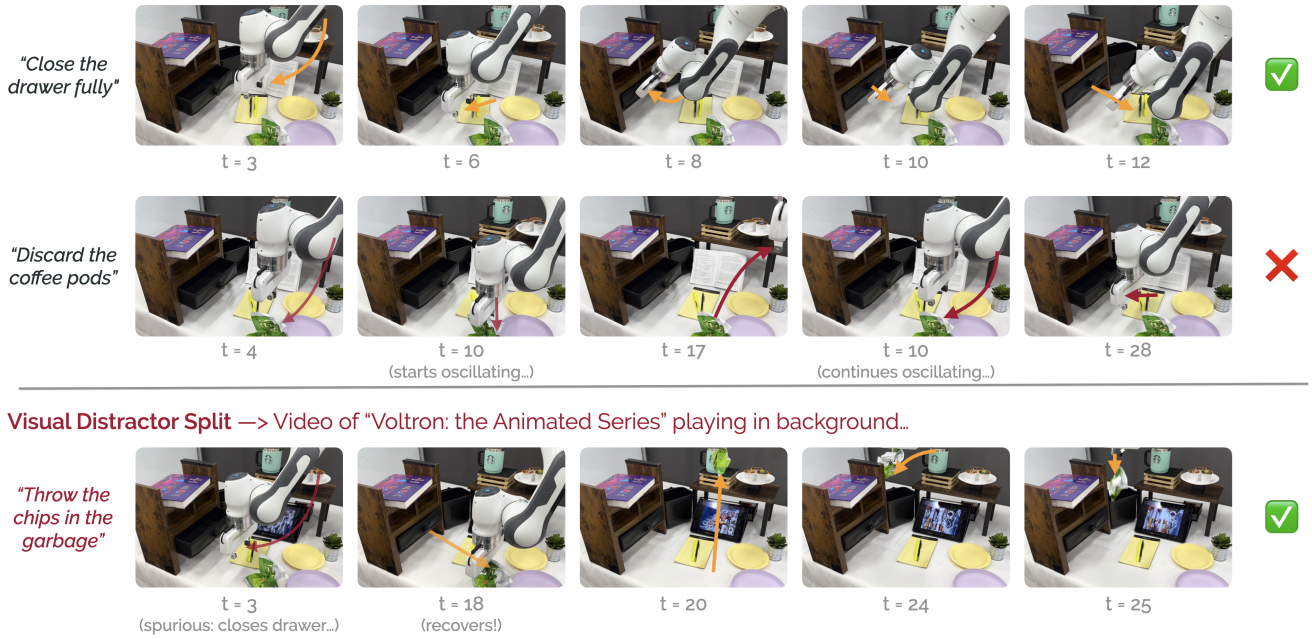

Figure 15. Real-World Language-Conditioned Imitation Rollouts from $\mathcal{V}$-Gen. We visualize some rollouts from the best-performing real-world language-conditioned imitation learning model, $\mathcal{V}$**-Gen**. While some tasks – e.g., discarding the plate of used coffee pods in the trash – prove hard for all methods, $\mathcal{V}$**-Gen** shows smooth motion on a series of tasks, even when challenging visual distractors are present. Videos with evaluation rollouts for each method are on our project page.

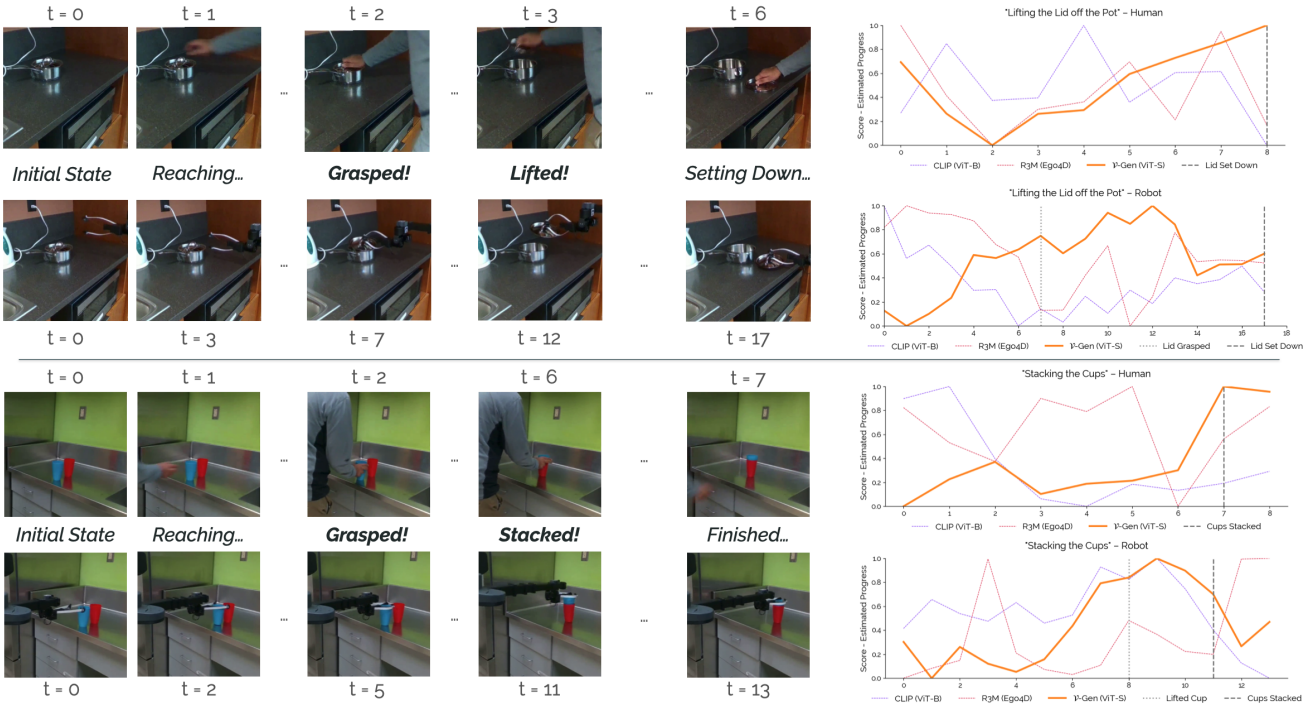

Figure 16. Additional Qualitative Zero-Shot Intent Scoring Examples. Given more videos of humans and robots performing similar behaviors from the WHiRL dataset (Bahl et al. 2022), we evaluate the zero-shot intent scoring capabilities of $\mathcal{V}$**-Gen**, *R3M (Ego4D)* and *CLIP (ViT-Base)*. In general, $\mathcal{V}$**-Gen** continues to show a nuanced understanding of semantics over time, in general tracking key points in each video smoothly, whereas both baselines are for the most part predicting random scores.

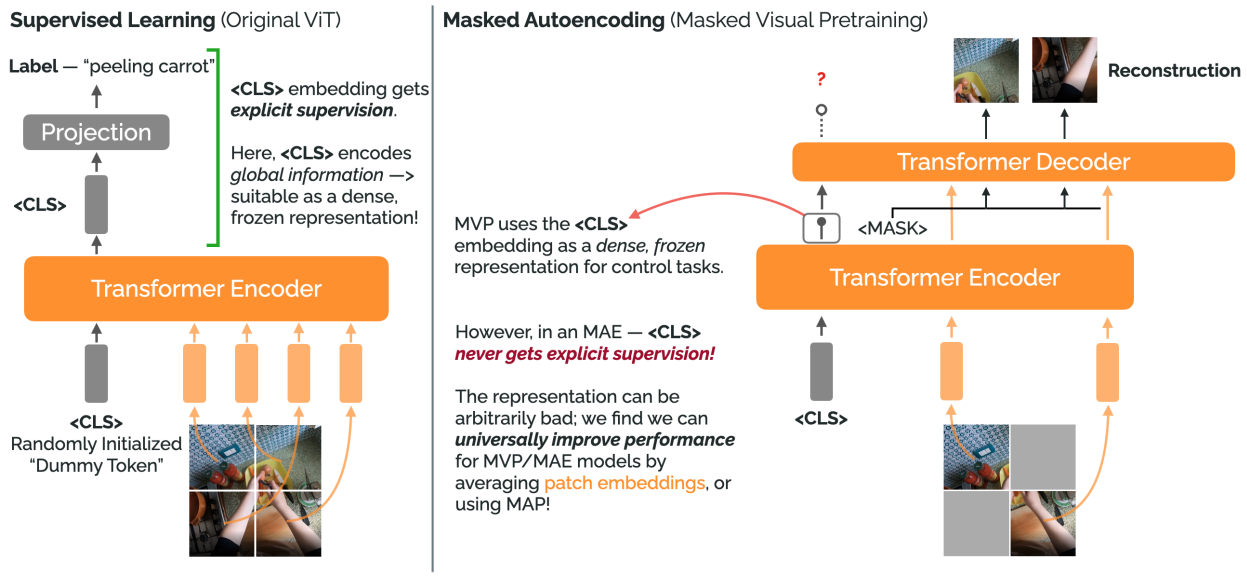

Figure 17. Default Feature Extraction in MAE Models. Prior work in masked autoencoding including MVP use the embedding corresponding to a dummy <CLS> token appended to the Transformer input for downstream adaptation. While this is motivated in the *supervised learning* setting, it is not clear what this embedding captures in the MAE setting, as it never receives explicit supervision. We find that pooling the *learned patch embeddings* is strictly better.

C Data-Equivalent Reproductions & Reproducibility

In this section we provide additional discussion around two aspects of the reproduction and pretraining procedure discussed in §4: 1) preprocessing, and specifically the *importance of selecting multiple images from the same context*, and 2) how to operationalize the representations from the visual encoder for downstream learning.

C.1 Additional Preprocessing Discussion

We described our preprocessing approach in §4: following the R3M paper, we sample *five frames from each video clip* for each epoch of pretraining. Seeing multiple frames from the same visual context is minimally necessary for the R3M time-contrastive learning objective, but we posit in this discussion that repeatedly sampling from the same visual context – even with a reconstruction objective – allows for picking up on finer-grained changes *within* a context. The best evidence we have for this is in looking at how prior work constructs their pretraining datasets.

The original MVP work (Xiao et al. 2022; Radosavovic et al. 2022) constructs *static datasets of images* by iterating through the various video clips in their pretraining datasets – Sth-Sth, Ego4D (Grauman et al. 2022), 100 Days of Hands (Shan et al. 2020) – at a fixed rate, usually from 0.2 to 1 frames per second. Given video clip lengths of 2 seconds, this means that *in aggregate* these pretraining datasets comprise maybe 2-3 frames sampled from the same clip, if that. Contrast that with this work and R3M, sampling multiple frames from *each video clip* for *every pretraining epoch* (for 400 epochs). This not only means that we are seeing the same context repeatedly, but also that we are seeing different *views* of the same context; this can help tune reconstruction towards picking up on finer-grained features (e.g., if a high-capacity model is able to memorize prior contexts given enough repetition).

This offers a speculative explanation of why Voltron models outperform *MVP (EgoSoup)* models that are both higher-capacity and trained on orders of magnitude more data – but definitely requires further experiments to prove. In the meantime, it seems as though taking steps to use as much of the pretraining datasets we have access to as possible is in our best interest.

C.2 Multiheaded Attention Pooling – Extracting Representations

There is a critical difference between pretraining visual representations and identifying the "right" way to use these representations for downstream adaptation tasks. Especially for Vision Transformers trained as part of a masked autoencoder – as mentioned at the end of Section §4 of the main text – identifying a method for extracting information from the learned representations is an open problem. The main text states – by fiat – that we use multiheaded attention pooling (MAP; Lee et al. 2018) as suggested by Zhai et al. (2022) to operationalize our learned representations for our downstream tasks. Here, we further contextualize that decision with a description of alternative approaches, as well as comparative results (Table 6) that show the superiority of MAP-based "feature extraction" (referring to the process of taking the output of a Vision Transformer and producing a dense, summary vector for downstream learning) over alternative approaches.

MVP and prior work in masked autoencoding with Vision Transformers (He et al. 2022) make an interesting choice when it comes to extracting features: during pretraining, these works append a dummy <CLS> token to the input of the encoder

Table 6. Feature Extraction Results. We evaluate various feature extraction strategies on the Franka Kitchen visuomotor control tasks at $n = 10$ demonstrations. We find that MAP is strictly superior for all Vision Transformer backbones; even mean-pooling over patch embeddings outperforms the default strategy from the MVP work that uses the frozen <CLS> embedding.

	Arch.	Default Extractor	Mean-Pooling	Attention Pooling (MAP)
R-R3M	ViT-S	16.07 (Default = Mean-Pooling)	–	14.73
R-MVP	ViT-S	7.90 (Default = <CLS> Token)	9.50	**26.73**
$\mathcal{V}$-Cond	ViT-S	–	19.07	**27.33**
$\mathcal{V}$-Dual	ViT-S	–	17.40	**33.07**
$\mathcal{V}$-Gen	ViT-S	–	15.67	**30.33**
$\mathcal{V}$-Cond	ViT-B	–	19.40	**30.80**
$\mathcal{V}$-Dual	ViT-B	–	16.40	**37.27**
$\mathcal{V}$-Gen	ViT-B	–	15.73	**32.13**
CLIP	ViT-B	17.73 (Default = Pool & Normalize)	16.33	**22.20**
MVP (EgoSoup)	ViT-B	18.20 (Default = <CLS>)	20.13	**33.87**

and decoder in the masked autoencoding pipeline (depicted in Fig. 17). This "free" embedding is motivated by how Vision Transformers for supervised learning (e.g., classification) are parameterized: in these settings, after encoding an input image, the <CLS> embedding is used as (the sole) input to a linear projection into label space, thus obtaining supervision from the global loss function (e.g., the cross-entropy loss for classification). Crucially, the <CLS> embedding in these cases gets *direct supervision* during training. However, in the masked autoencoding setting, this <CLS> embedding is just passed through the various Transformer layers of the encoder and decoder, *never obtaining any direct or indirect supervision*; while it does attend to all other patch embeddings as a byproduct of the multiheaded attention mechanism, there is no guarantee that this embedding captures or summarize all the useful information necessary.

Instead, recent work from the same authors of the original Vision Transformer (Zhai et al. 2022) eschew the <CLS> embedding completely during training, instead identifying that two other strategies – mean-pooling *all* the patch embeddings output by the encoder, or using multiheaded attention pooling (Lee et al. 2018) – are almost always preferable. As an aside – this work is what motivates Voltron models to also do away with the <CLS> embedding.

Multiheaded attention pooling (MAP) can be thought of as a form of cross-attention with a *learned* query. Starting with a randomly initialized query vector (or optionally, *set* of query vectors), a MAP block implements a shallow multiheaded attention operation, using the initialized query vector to cross-attend over the patch embeddings output by the Vision Transformer – the resulting output is a "weighted" combination of the individual patch embeddings that is shaped on a per-adaptation basis. We evaluate MAP-based extraction against mean-pooling and any other "default" strategy (e.g., the <CLS> embedding used in MVP, the learned dense representation under *CLIP*) in Table 6. We find that MAP universally outperforms all other strategies on the Franka Kitchen control tasks (with $n = 10$ demonstrations), informing our usage of MAP as the sole feature extraction approach throughout this work. Notably, we find that MAP-based extraction when applied to the original model *MVP (EgoSoup)* released in the original work *almost doubles success rate* on downstream control tasks. We even find that simple patch averaging outperforms the <CLS> embedding, further motivating alternatives.