IJCV Special Issue (Best Papers from ICCV 2015) manuscript No.
(will be inserted by the editor)

Learning image representations tied to egomotion from unlabeled video

Dinesh Jayaraman · Kristen Grauman

Received: 16 June 2016 / Accepted: Feb 10, 2017

Abstract Understanding how images of objects and scenes behave in response to specific egomotions is a crucial aspect of proper visual development, yet existing visual learning methods are conspicuously disconnected from the physical source of their images. We propose a new "embodied" visual learning paradigm, exploiting proprioceptive motor signals to train visual representations from egocentric video with no manual supervision. Specifically, we enforce that our learned features exhibit equivariance i.e., they respond predictably to transformations associated with distinct egomotions. With three datasets, we show that our unsupervised feature learning approach significantly outperforms previous approaches on visual recognition and next-best-view prediction tasks. In the most challenging test, we show that features learned from video captured on an autonomous driving platform improve large-scale scene recognition in static images from a disjoint domain.

1 Introduction

How is visual learning shaped by egomotion? In their famous "kitten carousel" experiment, psychologists Held and Hein examined this question in 1963 [18]. To analyze the role of self-produced movement in perceptual development, they designed a carousel-like apparatus in which two kittens could be harnessed. For eight weeks after birth, the kittens were kept in a dark environment, except for one hour a day on the carousel. One kitten, the "active" kitten, could move freely of its own volition while attached. The other kitten, the "passive"

D. Jayaraman
The University of Texas at Austin
E-mail: dineshj@cs.utexas.edu

K. Grauman
The University of Texas at Austin
E-mail: grauman@cs.utexas.edu

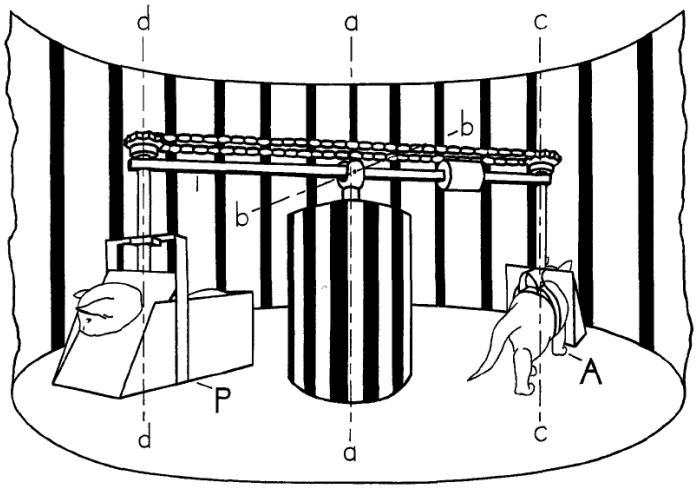

Fig. 1 Schematic figure from [18] showing the apparatus for the kitten carousel study. The active kitten 'A' was free to move itself in both directions around the three axes of rotation a-a, b-b and c-c, while pulling the passive kitten 'P' through the equivalent movements around a-a, b-b and d-d by means of the mechanical linkages in the carousel setup.

kitten, was carried along in a basket and could not control his own movement; rather, he was forced to move in exactly the same way as the active kitten. Fig 1 shows a schematic of the apparatus. Thus, both kittens had identical visual experiences. However, while the active kitten simultaneously experienced signals about his own motor actions, the passive kitten was simply along for the ride. It saw what the active kitten saw, but it could not simultaneously learn from self-generated motion signals.

The outcome of the experiment is remarkable. After eight weeks, the active kitten's visual perception was indistinguishable from kittens raised normally, whereas the passive kitten suffered fundamental problems. The implication is clear: proper perceptual development requires leveraging *self-generated movement in concert with visual feedback*. Specifically, the active kitten had two advantages over the passive kitten: (i) it had proprioceptive knowledge of the specific motions of its body that were causing the visual responses it was observing, and (ii) it had the ability to select those motions in the first place. The results of this experiment establish that these advantages are critical to the development of visual perception.

We contend that today's visual recognition algorithms are crippled much like the passive kitten. The culprit: learning from "bags of images". Ever since statistical learning methods emerged as the dominant paradigm in the recognition literature, the norm has been to treat images as i.i.d. draws from an underlying distribution. Whether learning object categories, scene classes, body poses, or features themselves, the idea is to discover patterns within a collection of snapshots, blind to their physical source. So is the answer to learn from video? Only partially. As we can see from the kitten carousel experiment, or

in general from observing biological perceptual systems, vision develops in the context of acting and moving in the world. Without leveraging the accompanying motor signals initiated by the observer, learning from video data does *not* escape the passive kitten's predicament.

Inspired by this concept, we propose to treat visual learning as an embodied process, where the visual experience is inextricably linked to the motor activity behind it.[1] In particular, our goal is to learn representations that exploit the parallel signals of egomotion and pixel appearance. As we will explain below, we hypothesize that downstream processing will benefit from access to such representations.

To this end, we attempt to learn the connection between how an observer moves, and how its visual surroundings change. We do this by exploiting motor signals accompanying unlabeled egocentric video, of the sort that one could obtain from a wearable platform like Google Glass, a self-driving car, or even a mobile phone camera.

To understand what we mean by learning the egomotion-vision connection, consider the "guess the new view" game, depicted in Fig 2(a). Given only one view of an object or a scene, the problem of computing what other views would look like is severely underdetermined. Yet, most often, humans are able to hallucinate such views. For instance, in the example of Fig 2(a), there are many hints in the first view that allow us to reasonably guess many aspects of the new view following the car's rotation. For instance, the traffic lights indicate that the observer must be at an intersection; the tree in the first view is probably closer to the camera than the tower, and will occlude the tower after the observer has moved; and it is even possible to extrapolate an entirely unseen face of the building using only geometric and semantic priors on the symmetry of buildings. The true view from the new position is shown in Fig 2(b).

We hypothesize that learning to solve this egomotion-conditioned view prediction task may help visual learning. As shown in the example above, view prediction draws on complex visual skills such as semantics (recognizing "building", "tree", "tower" etc.), depth and 3D geometry (the "tree" and the "tower"), and context ("traffic lights" $\Rightarrow$ "intersection"). These are general visual skills that are not limited to the view prediction task, but instead transfer well to many other tasks, including recognition. Moreoever, view prediction offers a way to acquire these skills entirely without manual labels.

In this work, we exploit this fact by incorporating the view prediction task above into an unsupervised *equivariant* feature learning approach using egocentric video and motor signals. During training, the input image sequences are accompanied by a synchronized stream of ego-motor sensor readings; however, they need not possess any semantic labels. The ego-motor signal could correspond, for example, to the inertial sensor measurements received alongside video on a wearable or car-mounted camera. The objective is to learn a

[1] Depending on the context, the motor activity could correspond to either the 6-DOF egomotion of the observer moving in the scene or the second-hand motion of an object being actively manipulated, e.g., by a person or robot's end effectors.

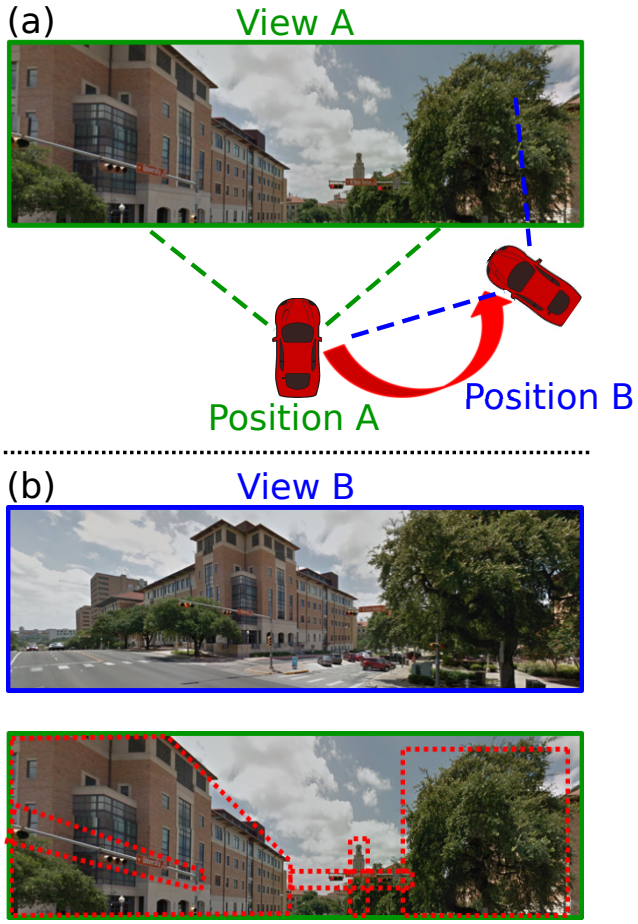

Fig. 2 Guess the new view: (a) Given the view, View A, out of the windshield of the car when in position A, can you guess what the view (View B) would look like, when the car shifts to position B? (b) View B, the answer to (a), can be reasonably guessed from semantic, geometric, depth and contextual cues from View A, as shown in red outlines below view B. See text for explanation.

feature mapping from pixels in a video frame to a space that is equivariant to various motion classes. In other words, the learned features should *change in predictable and systematic ways as a function of the transformation applied to the original input.* See Fig 3. We develop a convolutional neural network (CNN) approach that optimizes a feature map for the desired egomotion-based equivariance. We further show various ways in which this approach can be exploited for category learning — to produce input features to a generic classifier, to pretrain a network that is then finetuned for classification, or to regularize a classification loss. Egomotion thus serves as useful side information to guide the features learned, which we show facilitates category learning when labeled examples are scarce.

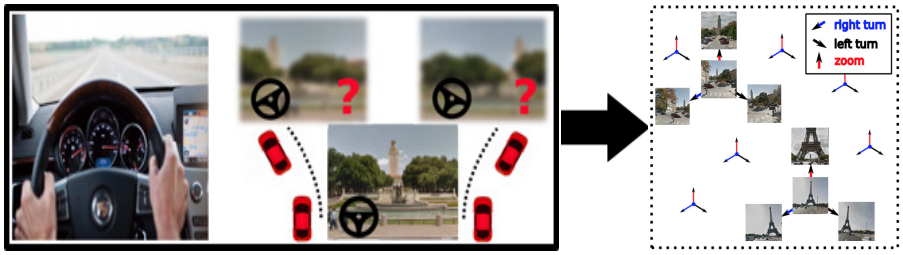

Fig. 3 Our approach learns an image embedding from unlabeled video. Starting from egocentric video together with observer egomotion signals, we train a system on a "view prediction" task (left), to learn *equivariant* visual features that respond predictably to observer egomotion (right). In this target equivariant feature space, pairs of images related by the same egomotion are related by the same *feature transformation* too.

In sharp contrast to our idea, previous work on visual features—whether hand-designed or learned—primarily targets feature *invariance*. Invariance is a special case of equivariance, where transformations of the input have no effect. Typically, one seeks invariance to small transformations, e.g., the orientation binning and pooling operations in SIFT/HOG [34,8,46] and modern CNNs both target invariance to local translations and rotations. While a powerful concept, invariant representations require a delicate balance: "too much" invariance leads to a loss of useful information or discriminability. In contrast, more general equivariant representations are intriguing for their capacity to impose structure on the output space without forcing a loss of information. Equivariance is "active" in that it exploits observer motor signals, thus partially modeling the advantages of Hein and Held's active kitten.

Our main contribution is a novel feature learning approach that couples ego-motor signals and unlabeled video. To our knowledge, ours is the first attempt to ground feature learning in physical activity. The limited prior work on unsupervised feature learning with video [37,39,36,16,48] learns only passively from observed scene dynamics, uninformed by explicit motor sensory cues. Furthermore, while equivariance is explored in some recent work, unlike our idea, it typically focuses on 2D image transformations as opposed to 3D egomotion [25,41] and considers existing features [46,30]. Finally, whereas existing methods that learn from image transformations focus on view synthesis applications [19,28,36], we explore recognition applications of learning jointly equivariant and discriminative feature maps.

We apply our approach to three public datasets. On pure equivariance as well as recognition tasks, our method consistently outperforms the most related techniques in feature learning. In the most challenging test of our method, we show that features learned from video captured on a vehicle can improve image recognition accuracy on a disjoint domain. In particular, we use unlabeled KITTI [13,14] car data for unsupervised feature learning for the 397-class scene recognition task for the SUN dataset [52]. Our results show the promise of departing from the "bag of images" mindset, in favor of an embodied approach to feature learning.

2 Related work

Invariant features Invariance is a special case of equivariance, wherein a transformed output remains identical to its input. Invariance is known to be valuable for visual representations. Descriptors like SIFT [34], HOG [8], and aspects of CNNs like pooling and convolution, are hand-designed for invariance to small shifts and rotations. Feature learning work aims to *learn* invariances from data [42,43,47,44,11]. Strategies include augmenting training data by perturbing image instances with label-preserving transformations [43,47,11], and inserting linear transformation operators into the feature learning algorithm [44].

Most relevant to our work are feature learning methods based on temporal coherence and "slow feature analysis" [50,17,37]. The idea is to require that learned features vary slowly over continuous video, since visual stimuli can only gradually change between adjacent frames. Temporal coherence has been explored for unsupervised feature learning with CNNs [37,55,16,5,33,48,12], with applications to dimensionality reduction [17], object recognition [37,55], and metric learning [16]. Temporal coherence of inferred body poses in unlabeled video is exploited for invariant recognition in [6]. These methods exploit video as a source of free supervision to achieve invariance, analogous to the image perturbations idea above. In contrast, our method exploits video coupled with ego-motor signals to achieve the more general property of equivariance.

Equivariant representations Equivariant features can also be hand-designed or learned. For example, equivariant or "co-variant" operators are designed to detect repeatable interest points [46]. Recent work explores ways to learn descriptors with in-plane translation/rotation equivariance [25,41]. While the latter does perform feature learning, its equivariance properties are crafted for specific 2D image transformations. In contrast, we target more complex equivariances arising from natural observer motions (3D egomotion) that cannot easily be crafted, and our method learns them from data.

Methods to learn representations with disentangled latent factors [19,28] aim to sort properties like pose and illumination into distinct portions of the feature space. For example, the transforming auto-encoder learns to explicitly represent instantiation parameters of object parts in equivariant hidden layer units [19]. Such methods target equivariance in the limited sense of inferring pose parameters, which are appended to a conventional feature space designed to be invariant. In contrast, our formulation encourages equivariance over the *complete* feature space; we show the impact as an unsupervised regularizer when training a recognition model with limited training data.

It has been shown to be possible to predict poses of objects using descriptors learned for classification tasks [45]. The work of [30] quantifies the invariance/equivariance of various standard representations, including CNN features, in terms of their responses to specified in-plane 2D image transformations (affine warps, flips of the image). We adopt the definition of equivariance used in that work, but our goal is entirely different. While these works demon-

strate and/or quantify the equivariance of existing descriptors, our approach focuses on learning a feature space that is equivariant.

Learning transformations Other methods train with pairs of transformed images and infer an implicit representation for the transformation itself. In [35], bilinear models with multiplicative interactions are used to learn content-independent "motion features" that encode only the transformation between image pairs. One such model, the "gated autoencoder" is extended to perform sequence prediction for video in [36]. Recurrent neural networks combined with a grammar model of scene dynamics can also predict future frames in video [39]. Whereas these methods learn a representation for image pairs (or tuples) related by some transformation, we learn a representation for individual images in which the behavior under transformations is predictable. Furthermore, whereas these prior methods abstract away the image content, our method preserves it, making our features relevant for recognition.

Egocentric vision There is renewed interest in egocentric computer vision methods, though none perform feature learning using motor signals and pixels in concert as we propose. Recent methods use egomotion cues to separate foreground and background [40,53] or infer the first-person gaze [54,32]. While most work relies solely on apparent image motion, the method of [53] exploits a robot's motor signals to detect moving objects and [38] uses reinforcement learning to form robot movement policies by exploiting correlations between motor commands and observed motion cues.

Vision from/for motion Very recently, concurrently with our work, and independent of it, a growing body of work [7,26,49,2,3,31] studies the interaction between high-level visual tasks and agent actions or motions. Among these, [31, 26,49,3] focus on end-to-end learning of visual representations targeting action tasks such as driving. Some work also studies the theoretical properties of visual representations that vary linearly with observer motion [7], a form of equivariance, or learns a visual representation space in which control tasks simplify to linear operations [49]. Of all these recent works, the closest to ours is [2], which uses a different approach to ours to learn visual representations from video with associated egomotion sensor streams. Rather than learn to predict a new view given the starting view and the egomotion as we do, their method learns to predict the *egomotion*, given the original and final views. Conceptually, while our approach explicitly targets a desired property, egomotion-equivariance, in the learned feature space, the method of [2] treats their egomotion-regression task as a generic proxy task for representation learning. We compare against their method in our experiments in Sec 4.

Finally, this manuscript builds upon our previous work published in ICCV 2015 [21]. Specifically, we make the following additional contributions in this work: (i) we conceptually extend our equivariant feature learning formulation to handle non-discrete motions and more general definitions of equivariance

(Sec 3.4), (ii) we empirically verify that our method scales up to much larger images than in previous tests (Sec 4.4), (iii) we study the impact of the equivariance objective on multiple layers of features in a deep neural network architecture (Sec 4.4), (iv) we show that features trained purely for equivariance in our formulation, entirely without manual supervision, may be used as inputs to a generic classifier for recognition tasks (Sec 4.4), (v) we empirically verify that the network weights corresponding to such purely unsupervised equivariant features may be finetuned for recognition tasks (Sec 4.4), (vi) we perform new experiments allowing the direct comparison of features learned in a neural network classifier with unsupervised egomotion-equivariance regularization, and features trained purely for egomotion-equivariance (Sec 4.5) (vii) we present alternative intuitions supporting our equivariance formulation in terms of new view prediction (Sec 1), (viii) we compare our approach against a new baseline, LSM [2], and (ix) we significantly extend all sections of the paper, including our descriptions of the method, its motivations and experiments, with additional illustrations for the sake of clarity and completeness.

3 Approach

Our goal is to learn an image representation that is equivariant with respect to egomotion transformations. Let $\boldsymbol{x}_i \in \mathcal{X}$ be an image in the original pixel space, and let $\boldsymbol{y}_i \in \mathcal{Y}$ be its associated ego-pose representation. The ego-pose captures the available motor signals, and could take a variety of forms. For example, $\mathcal{Y}$ may encode the complete observer camera pose (its position in 3D space, pitch, yaw, roll), some subset of those parameters, or any reading from a motor sensor paired with the camera.

As input to our learning algorithm, we have a training set $\mathcal{U}$ of N_u unlabeled image pairs and their associated ego-poses, $\mathcal{U} = \{\langle(\boldsymbol{x}_i, \boldsymbol{x}_j), (\boldsymbol{y}_i, \boldsymbol{y}_j)\rangle\}_{(i,j)=1}^{N_u}\}$. The image pairs originate from video sequences, though they need not be adjacent frames in time. The set may contain pairs from multiple videos and cameras. Note that this training data does *not* have any semantic labels (object categories, etc.); they are "labeled" only in terms of the ego-motor sensor readings. Since our method relies on freely available motion sensor readings associated with video streams (e.g., from Google glass, self-driving cars, or even hand-held mobile devices), rather than on expensive manually supplied labels, it is effectively unsupervised.[2]

In the following, we first explain how to translate ego-pose information into pairwise "motion pattern" annotations (Sec 3.1). Then, Sec 3.2 defines the precise nature of the equivariance we seek, and Sec 3.3 defines our learning

[2] One could attempt to apply our idea using camera poses inferred from the video itself (i.e., with structure from motion). However, there are conceptual and practical advantages to relying instead on external sensor data capturing egomotion. First, the sensor data, when available, is much more efficient to obtain and can be more reliable. Second, the use of an external sensor parallels the desired effect of the agent learning from its proprioception motor signals, as opposed to bootstrapping the visual learning process from a previously defined visual odometry module based on the same visual input stream.

objective. We define a variant of our approach using non-discrete egomotion patterns and non-linear equivariance maps in Sec 3.4. Then, in Sec 3.5, we show how a feedforward neural network architecture may be trained to produce the desired equivariant feature space. Finally, Sec 3.6 shows how our equivariant feature learning scheme may be used to enhance recognition with limited training data.

3.1 Mining discrete egomotion patterns

First we want to organize training sample pairs into a discrete set of egomotion patterns $\mathcal{G}$. For instance, one egomotion pattern might correspond to "tilt downwards by approximately 20°". As we will see in Sec 3.3, translating raw egomotion signals into a few discrete motion patterns helps to simplify the design of our system. While one could collect new data explicitly controlling for the patterns (e.g., with a turntable and camera rig), we prefer a data-driven approach that can leverage video and ego-pose data collected "in the wild".

To this end, we discover clusters among pose difference vectors $\boldsymbol{y}_i - \boldsymbol{y}_j$ for pairs (i, j) of temporally close frames from video (typically less than 1 second apart; see Sec 4.1 for details). For simplicity we apply k-means to find G clusters, though other methods are possible. Let $p_{ij} \in \mathcal{P} = \{1, \ldots, G\}$ denote the motion pattern ID, i.e., the cluster to which $(\boldsymbol{y}_i, \boldsymbol{y}_j)$ belongs. We can now replace the ego-pose vectors in $\mathcal{U}$ with motion pattern IDs: $\langle(\boldsymbol{x}_i, \boldsymbol{x}_j), p_{ij}\rangle$. [3]

Fig 4 illustrates motion pattern discovery on frame pairs from the KITTI dataset [13,14] videos, which are captured from a moving car. Here $\mathcal{Y}$ consists of the position and yaw angle of the camera. So, we are clustering a 2D space consisting of forward distance and change in yaw. As shown in the bottom panel, the largest clusters correspond to the car's three primary egomotions: turning left, turning right, and going forward.

In Sec 3.4 we discuss a variant of our approach that operates with non-discrete motion patterns.

3.2 Definition of egomotion equivariance

Given $\mathcal{U}$, we wish to learn a feature mapping function $\mathbf{z}_{\boldsymbol{\theta}}(.) : \mathcal{X} \rightarrow \mathcal{R}^D$ parameterized by $\boldsymbol{\theta}$ that maps a single image to a D-dimensional vector space that is equivariant to egomotion.

To define equivariance, it is convenient to start with the notion of feature invariance, which is the standard property that visual representations for recognition are designed to exhibit. Invariant features are *unresponsive* to certain classes of so-called "nuisance transformations" such as observer egomotions, pose change, or illumination change. For images $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ with

[3] For movement with d degrees of freedom, setting $G \approx d$ should suffice (cf. Sec 3.2). Sec 3.3 discusses tradeoffs involved in selecting G. We chose a small value for G for efficiency and did not vary it in experiments.

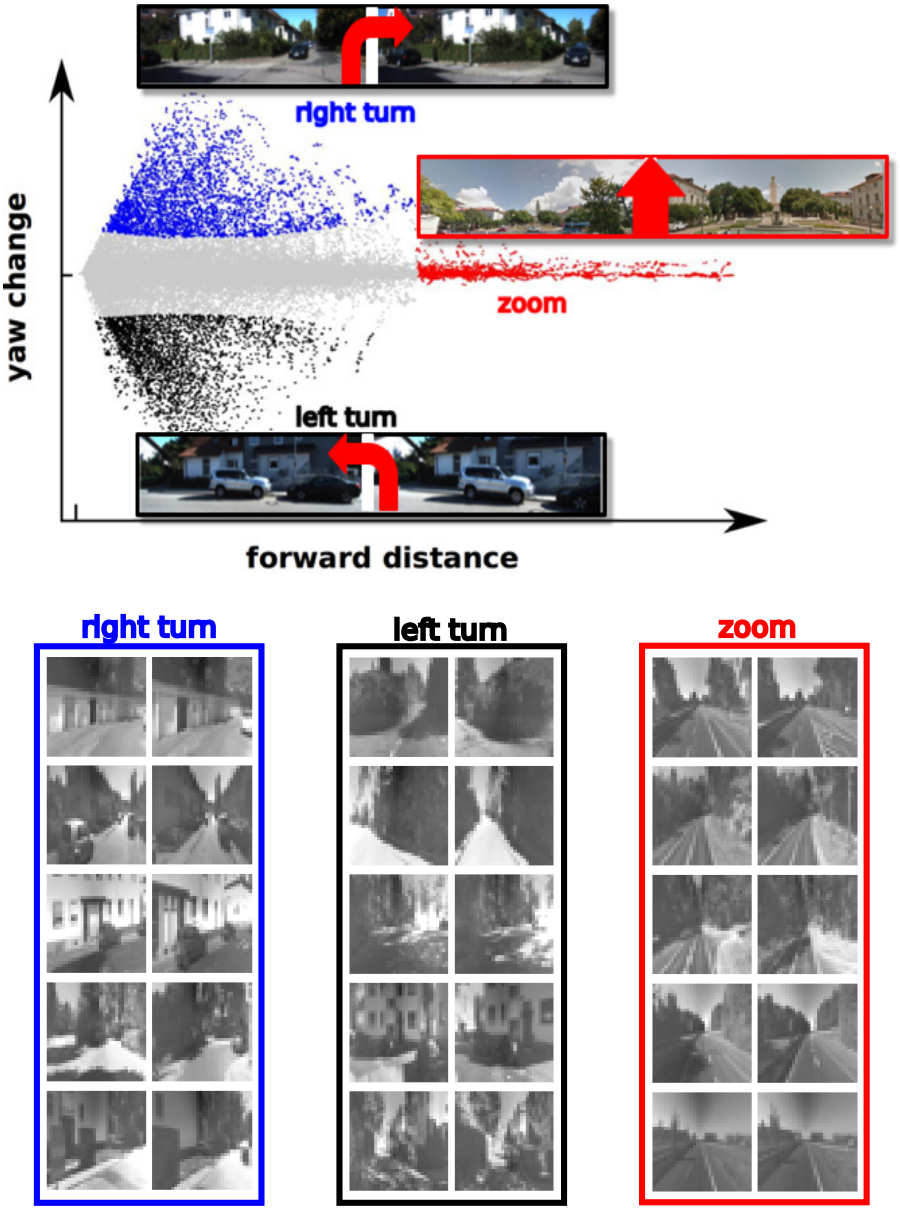

Fig. 4 Motion pattern discovery in KITTI car-mounted videos. (Top) Largest motion clusters in the "forward distance"-"yaw" space correspond to forward motion or "zoom", "right turn" and "left turn" respectively. (Bottom) Some example pairs corresponding to discovered motion patterns. Within each box corresponding to one motion pattern, each row corresponds to a pair.

associated ego-poses $\boldsymbol{y}_i$ and $\boldsymbol{y}_j$ respectively, an egomotion-invariant feature mapping function $\mathbf{z}_{\boldsymbol{\theta}}$ satisfies:

$$\mathbf{z}_{\boldsymbol{\theta}}(\boldsymbol{x}_j) \approx \mathbf{z}_{\boldsymbol{\theta}}(\boldsymbol{x}_i). \tag{1}$$

Recall that this is the form representation sought by many existing feature learning methods, including those that learn representations from video [37, 55, 16, 5, 33, 48, 12].

Rather than being *unresponsive* as above, equivariant functions are *predictably* responsive to transformations, i.e., an egomotion-equivariant function $\mathbf{z}_{\boldsymbol{\theta}}$ must respond systematically and predictably to egomotions:

$$\mathbf{z}_{\boldsymbol{\theta}}(\boldsymbol{x}_j) \approx f(\mathbf{z}_{\boldsymbol{\theta}}(\boldsymbol{x}_i), \boldsymbol{y}_i, \boldsymbol{y}_j), \tag{2}$$

for some simple function $f \in \mathcal{F}$, where again $\boldsymbol{y}_i$ denotes the ego-pose metadata associated with video frame $\boldsymbol{x}_i$. Note that f must be *simple*; as the space of allowed functions $\mathcal{F}$ grows larger, the requirement in Eq (2) above is satisfied by more feature mapping functions $\mathbf{z}_{\boldsymbol{\theta}}$. In other words, as $\mathcal{F}$ grows large, the equivariance constraint on $\mathbf{z}_{\boldsymbol{\theta}}$ grows weak.

We will first consider equivariance for *linear* functions $f(.)$, following [30]. Later, in Sec 3.4, we will show how to extend this to the non-linear case. In the linear case, $\mathbf{z}_{\boldsymbol{\theta}}$ is said to be equivariant with respect to some transformation g if there exists a $D \times D$ matrix[4] M_g such that:

$$\forall \boldsymbol{x} \in \mathcal{X} : \mathbf{z}_{\boldsymbol{\theta}}(g\boldsymbol{x}) \approx M_g \mathbf{z}_{\boldsymbol{\theta}}(\boldsymbol{x}). \tag{3}$$

Such an M_g is called the "equivariance map" of g on the feature space $\mathbf{z}_{\boldsymbol{\theta}}(.)$. It represents the affine transformation in the feature space that corresponds to transformation g in the pixel space. For example, suppose a motion pattern g corresponds to a yaw turn of 20°, and $\boldsymbol{x}$ and $g\boldsymbol{x}$ are the images observed before and after the turn, respectively. Equivariance demands that there is some matrix M_g that maps the pre-turn image to the post-turn image, once those images are expressed in the feature space $\mathbf{z}_{\boldsymbol{\theta}}$. Hence, $\mathbf{z}_{\boldsymbol{\theta}}$ "organizes" the feature space in such a way that movement in a particular direction in the feature space (here, as computed by matrix-vector multiplication with M_g) has a predictable outcome. The linear case, as also studied in [30], ensures that the structure of the mapping has a simple form — the space $\mathcal{F}$ of possible equivariance maps is suitably restricted so that the equivariance constraint is significant, as discussed above. It is also convenient for learning since M_g can be encoded as a fully connected layer in a neural network. In Sec 4, we experiment with both linear and simple non-linear equivariance maps.

3.2.1 Equivariance in dynamic 3D scenes

While prior work [25, 41] focuses on equivariance where g is a 2D image warp, we explore the case where $g \in \mathcal{P}$ is an egomotion pattern (cf. Sec 3.1) reflecting

[4] bias dimension assumed to be included in D for notational simplicity

the observer's 3D movement in the world. In theory, appearance changes of an image in response to an observer's egomotion are not determined completely by the egomotion alone. They also depend on the depth map of the scene and the motion of dynamic objects in the scene. One could easily augment either the frames $\boldsymbol{x}_i$ or the ego-pose $\boldsymbol{y}_i$ with depth maps, when available. Non-observer motion appears more difficult, especially in the face of changing occlusions and newly appearing objects. Even accounting for everything, a future frame may never be fully predictable purely from egomotion alone, due to changing occlusions/ newly visible elements in the scene. However, our experiments indicate we can learn effective representations even with dynamic objects and changing occlusions. In our implementation, we train with pairs relatively close in time, so as to avoid some of these pitfalls.

3.2.2 Equivariance to composite motions

While during training we target equivariance for the discrete set of G egomotions, if we use linear equivariance maps as above, the learned feature space will *not* be limited to preserving equivariance for pairs originating from the same egomotions. This is because the linear equivariance maps are composable. If we are operating in a space where every egomotion can be composed as a sequence of "atomic" motions, equivariance to those atomic motions is sufficient to guarantee equivariance to all motions.

To see this, suppose that the maps for "turn head right by 10°" (egomotion pattern r) and "turn head up by 10°" (egomotion pattern u) are respectively M_r and M_u, i.e, $\mathbf{z}(r\boldsymbol{x}) = M_r\mathbf{z}(\boldsymbol{x})$ and $\mathbf{z}(u\boldsymbol{x}) = M_u\mathbf{z}(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathcal{X}$. Now for a novel diagonal motion d that can be composed from these atomic motions as $d = r \circ u$ ("turn head up by 10°, then right by 10°"), we have:

$$\begin{aligned} \mathbf{z}(d\boldsymbol{x}) &= \mathbf{z}((r \circ u)\boldsymbol{x}) \\ &= M_r\mathbf{z}(u\boldsymbol{x}) \\ &= M_rM_u\mathbf{z}(\boldsymbol{x}), \end{aligned} \tag{4}$$

so that, setting $M_d := M_rM_u$, we have:

$$\mathbf{z}(d\boldsymbol{x}) = M_d\mathbf{z}(\boldsymbol{x}). \tag{5}$$

Comparing this against the definition of equivariance in Eq (3), we see that $M_d = M_rM_u$ is the equivariance map for the novel egomotion $d = r \circ u$, even though d was not among $1, \ldots, G$. This property lets us restrict our attention to a relatively small number of discrete egomotion patterns during training, and still learn features equivariant with respect to new egomotions. Sec 3.4 presents a variant of our method that operates without discretizing egomotions.

3.3 Equivariant feature learning objective

We now design a loss function that encourages the learned feature space $\mathbf{z}_{\boldsymbol{\theta}}$ to exhibit equivariance with respect to each egomotion pattern. Specifically, we would like to learn the optimal feature space parameters $\boldsymbol{\theta}^*$ jointly with its equivariance maps $\mathcal{M}^* = \{M_1^*, \ldots, M_G^*\}$ for the motion pattern clusters 1 through G (cf. Sec 3.1).

To achieve this, a naive translation of the definition of equivariance in Eq (3) into a minimization problem over feature space parameters $\boldsymbol{\theta}$ and the $D \times D$ equivariance map candidate matrices $\mathcal{M}$ (assuming linear maps) would be as follows:

$$(\boldsymbol{\theta}^*, \mathcal{M}^*) = \arg\min_{\boldsymbol{\theta}, \mathcal{M}} \sum_{g} \sum_{\{(i,j):p_{ij}=g\}} d\left(M_g \mathbf{z}_{\boldsymbol{\theta}}(\boldsymbol{x}_i), \mathbf{z}_{\boldsymbol{\theta}}(\boldsymbol{x}_j)\right), \tag{6}$$

where $d(.,.)$ is a distance measure. This problem can be decomposed into G independent optimization problems, one for each motion, corresponding only to the inner summation above, and dealing with disjoint data. The g-th such problem requires only that training frame pairs annotated with motion pattern $p_{ij} = g$ approximately satisfy Eq (3).

However, such a formulation admits problematic solutions that perfectly optimize it. For example, for the trivial all-zero feature space $\mathbf{z}_{\boldsymbol{\theta}}(\boldsymbol{x}) = \mathbf{0}, \forall \boldsymbol{x} \in \mathcal{X}$ with M_g set to the all-zeros matrix for all g, the loss above evaluates to zero. To avoid such solutions, and to force the learned M_g's to be different from one another (since we would like the learned representation to respond *differently* to different egomotions), we simultaneously account for the "negatives" of each motion pattern. Our learning objective is:

$$(\boldsymbol{\theta}^*, \mathcal{M}^*) = \arg\min_{\boldsymbol{\theta}, \mathcal{M}} \sum_{g,i,j} d_g\left(M_g \mathbf{z}_{\boldsymbol{\theta}}(\boldsymbol{x}_i), \mathbf{z}_{\boldsymbol{\theta}}(\boldsymbol{x}_j), p_{ij}\right), \tag{7}$$

where $d_g(.,.,.)$ is a "contrastive loss" [17] specific to motion pattern g:

$$d_g(\boldsymbol{a}, \boldsymbol{b}, c) = \mathbb{1}(c = g) d(\boldsymbol{a}, \boldsymbol{b}) + \mathbb{1}(c \neq g) \max(\delta - d(\boldsymbol{a}, \boldsymbol{b}), 0), \tag{8}$$

where $\mathbb{1}(.)$ is the indicator function. This contrastive loss penalizes distance between $\boldsymbol{a}$ and $\boldsymbol{b}$ in "positive" mode (when $c = g$), and pushes apart pairs in "negative" mode (when $c \neq g$), up to a minimum margin distance specified by the constant δ. We use the ℓ_2 norm for the distance $d(.,.)$.

In our objective in Eq (7), the contrastive loss operates in the latent feature space. For pairs belonging to cluster g, the contrastive loss d_g penalizes feature space distance between the first image and its transformed pair, similar to Eq (6) above. For pairs belonging to clusters other than g, the loss d_g requires that the transformation defined by M_g must not bring the image representations close together. In this way, our objective learns the M_g's jointly. It ensures that distinct egomotions, when applied to an input $\mathbf{z}_{\boldsymbol{\theta}}(\boldsymbol{x})$, map it to different locations in feature space. We discuss how the feature mapping function parameters are optimized below in Sec 3.5.

Note that the objective of Eq (8) depends on the choice G of the number of discovered egomotion patterns from Sec 3.1. As remarked earlier, for movement with d degrees of freedom, setting $G \approx d$ should suffice (cf. Sec 3.2). There are several tradeoffs involved in selecting G: (i) The more the clusters, the fewer the training samples in each. This could lead to overfitting of equivariance maps M_g, so that optimizing Eq (8) may no longer produce truly equivariant features. (ii) The more the clusters, the more the number of parameters to be held in memory during training — each cluster has a corresponding equivariance map module. (iii) The fewer the clusters, the more noisy the training sample labels. Fewer clusters lead to larger clusters with more lossy quantization of egomotions in the training data. This might adversely affect the quality of training. In practice, for our experiments in Sec 4, we observed that this dependence on G is not a problem — a small value for G is both efficient and produces good features. We did not vary G in experiments.

We now highlight the important distinctions between our objective of Eq (8) and the "temporal coherence" objective of [37], which is representative of works learning representations from video through slow feature analysis [55, 16, 5, 33, 48, 12]. Written in our notation, the objective of [37] may be stated as:

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \sum_{i,j} d_1(\mathbf{z}_{\boldsymbol{\theta}}(\boldsymbol{x}_i), \mathbf{z}_{\boldsymbol{\theta}}(\boldsymbol{x}_j), \mathbb{1}(|t_i - t_j| \leq T)), \tag{9}$$

where t_i, t_j are the video time indices of $\boldsymbol{x}_i$, $\boldsymbol{x}_j$ and T is a temporal neighborhood size hyperparameter. This loss encourages the representations of nearby frames to be similar to one another, learning invariant representations. To see this, note how this loss directly optimizes representations to exhibit the invariance property defined in Eq (1). However, crucially, it does not account for the nature of the egomotion between the frames. Accordingly, while temporal coherence helps learn invariance to small image changes, it does not target a (more general) equivariant space. Like the passive kitten from Hein and Held's experiment, the temporal coherence constraint watches video to passively learn a representation; like the active kitten, our method registers the *observer motion* explicitly with the video to learn more effectively, as we will demonstrate in results.

3.4 Equivariance in non-discrete motion spaces with non-linear equivariance maps

Thus far, we have dealt with a discrete set of motions $\mathcal{G}$. When using linear equivariance maps, due to the composability of the maps, equivariance to all motions is guaranteed by equivariance to only the discrete set of motions in $\mathcal{G}$, so long as those discrete motions span the full motion space (Sec 3.2).

Still, the discrete motion solution has two limitations. First, it only generalizes to all egomotions for the restricted notion of equivariance relying on

linear maps, defined in Eq (3). In particular, for non-linear equivariance mapping functions $f(.)$ in the more general definition of equivariance in Eq (2), it does not guarantee equivariance to all egomotions. While linear maps nonetheless may be preferable for injecting stronger equivariance regularization effects, it is worth considering more general function families. Secondly, it is lossy, as it requires discretizing the continuous space of all motions into specific clusters. More specifically, image pairs assigned to the same cluster may be related by slightly different observer motions. This information is necessarily ignored by this motion discretization solution.

On the other hand, directly learning an infinite number of equivariance maps M_g, one corresponding to each motion g in the training set, is intractable. In this section, we develop a variant of our approach that implicitly learns these infinite equivariance maps and allows it to naturally transcend the linearity constraint on equivariance maps.

We now describe this non-discrete variant of our method. The set of egomotions $\mathcal{G}$ may now be an infinite, uncountable set of motions. As an example, we will assume the set of all motions in the training set:

$$\mathcal{G} = \{\boldsymbol{y}_i - \boldsymbol{y}_j;\ i, j \ \text{ are temporally nearby frames in training video}\}, \quad (10)$$

where $\boldsymbol{y}_i$ is the ego-pose associated with frame $\boldsymbol{x}_i$, as defined before.

Now, rather than attempting to learn separate equivariant maps M_g for each motion $g \in \mathcal{G}$, we may parameterize the entire family of M_g's through a single matrix function $\mathbf{M}$, as: $M_g = \mathbf{M}(g)$. Substituting this in Eq (3), we now want:

$$\mathbf{z}_{\boldsymbol{\theta}}(\boldsymbol{x}_j) \approx \mathbf{M}(\boldsymbol{y}_i - \boldsymbol{y}_j)\mathbf{z}_{\boldsymbol{\theta}}(\boldsymbol{x}_i). \quad (11)$$

At a high level, this may be thought of as similar to forming $G = \infty$ egomotion clusters for use with the discrete egomotions approach developed above (or more precisely, as many clusters as the number of egomotion-labeled training pairs i.e., $G = N_u$). Until this stage, our equivariance maps remain linear, as in Eq (3). However, since we are no longer restricted to a discrete set of motions $\mathcal{G}$, we need no longer rely on the composability of linear equivariance maps. Instead, we can further generalize our maps as follows:

$$\mathbf{z}_{\boldsymbol{\theta}}(\boldsymbol{x}_j) \approx \mathbf{M}(\mathbf{z}_{\boldsymbol{\theta}}(\boldsymbol{x}_i), \boldsymbol{y}_i - \boldsymbol{y}_j), \quad (12)$$

where $\mathbf{M}$ is now a function that produces a vector in the learned feature space as output. Note how this compares against the general notion of equivariance first defined in Eq (2).

Our general "non-discrete" equivariance objective may now be stated as:

$$(\boldsymbol{\theta}^*, \mathbf{M}^*) = \arg\min_{\boldsymbol{\theta}, \mathbf{M}} \sum_{i,j} d\left(\mathbf{M}(\mathbf{z}_{\boldsymbol{\theta}}(\boldsymbol{x}_i), \boldsymbol{y}_i - \boldsymbol{y}_j), \mathbf{z}_{\boldsymbol{\theta}}(\boldsymbol{x}_j)\right), \quad (13)$$

where $d(.,.)$ is a distance measure. Note that the objective in Eq (13) parallels the alternative objective in Eq (7) for the discrete motion case.[5] The architec-

[5] However, while the loss of Eq (7) is contrastive, Eq (13) specifies a non-contrastive loss. To overcome this deficiency in our experiments, we optimize this non-discrete equivariance loss only in conjunction with an auxiliary contrastive loss, such as DRLIM [17].

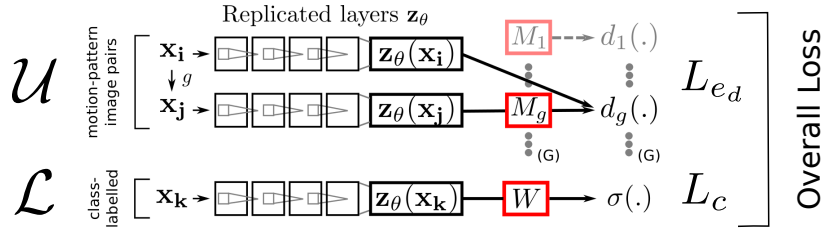

Fig. 5 Training setup for discrete egomotions: (top) a two-stack "Siamese network" processes video frame pairs identically before optimizing the equivariance loss of Eq (7), and (bottom) a third layer stack simultaneously processes class-labeled images to optimize the supervised recognition softmax loss as in Eq (14). See Sec 4.1 for exact network specifications.

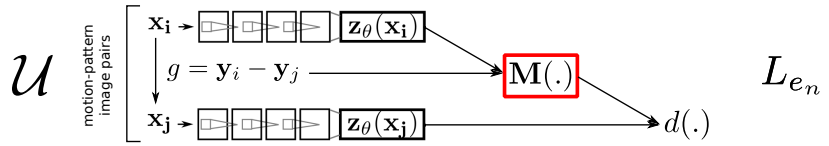

Fig. 6 Unsupervised training setup for the "non-discrete" variant Eq (13) of the equivariance objective. First, layer stacks with tied weights, representing the feature mapping $\mathbf{z}_{\boldsymbol{\theta}}$ to be learned, process video frame pairs identically to embed them into a feature space. In this space, an equivariance mapping function $M(.)$ acts on the first frame and camera egomotion vector g to attempt to predict the second frame. See Fig 7 for the architecture of $\mathbf{M}(.)$, and Sec 4.1 for further details. When used in a regularization setup with labeled data $\mathcal{L}$, a third stack of layers may be added as in Fig 5 to compute the classification loss.

ture of the function $\mathbf{M}(.)$ and how it is trained, are specified in Sec 3.5 and Sec 4.1.

This non-discretized motion and non-linear equivariance formulation allows an easy way to control the strength of the equivariance objective. The more complex the class of functions modeled by $\mathbf{M}(.)$, the weaker the notion of equivariance that is imposed upon the learned feature space. Moreover, it does not require discarding fine-grained information among the egomotion labels, as in the discrete motion case. We evaluate the impact of these conceptual differences, in experiments (Sec 4.4).

3.5 Form of the feature mapping function $\mathbf{z}_{\boldsymbol{\theta}}(.)$

For the mapping $\mathbf{z}_{\boldsymbol{\theta}}(.)$, we use a convolutional neural network architecture, so that the parameter vector $\boldsymbol{\theta}$ now represents the layer weights. We start with the discrete egomotions variant of our method. Let L_{e_d} denote the equivariance loss of Eq (7) based on discretized egomotions. L_{e_d} is optimized by sharing the weight parameters $\boldsymbol{\theta}$ among two identical stacks of layers in a "Siamese" network [4,17,37], as shown in the top two rows of Fig 5. Video frame pairs from $\mathcal{U}$ are fed into these two stacks. Both stacks are initialized with identical random weights, and identical gradients are passed through them in every

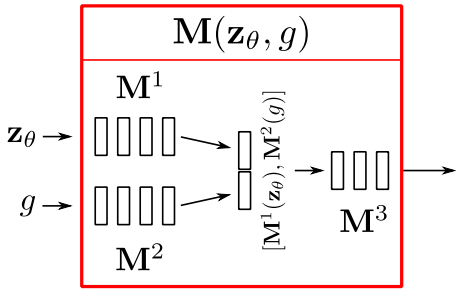

Fig. 7 Architecture of the module $\mathbf{M}(.)$ used for optimizing the non-discrete equivariance objective of Eq (13). The feature vector $\mathbf{z_\theta}$ and the continuous egomotion vector $g = \boldsymbol{y}_i - \boldsymbol{y}_j$ are processed through separate neural network modules $\mathbf{M}^1$ and $\mathbf{M}^2$ respectively, before appending and processing through a final module $\mathbf{M}^3$ that produces an output in the same domain as the input $\mathbf{z_\theta}$. Fig 6 shows where this fits into the full Siamese network framework.

training epoch, so that the weights remain tied throughout. Each stack encodes the feature map that we wish to train, $\mathbf{z_\theta}$.

To optimize Eq (7), an array of equivarance maps $\mathcal{M}$, each represented by a fully connected layer, is connected to the top of the second stack. Each such equivariance map then feeds into a motion-pattern-specific contrastive loss function d_g, whose other inputs are the first stack output and the egomotion pattern ID p_{ij}. This Siamese network architecture is depicted in the $\mathcal{U} \rightarrow L_{e_d}$ pipeline in Fig 5 (top).

Optimization is done through mini-batch stochastic gradient descent implemented through backpropagation with the Caffe package [24] (more details in Sec 4 and the Appendix).

For the case of the non-discretized motions variant of our approach in Eq (13), let L_{e_n} denote the equivariance loss of Eq (13). L_{e_n} is optimized as follows. The array of equivariance maps $\mathcal{M}$ is replaced by a single module $\mathbf{M}(\mathbf{z_\theta}, g)$, as shown in Fig 6. The architecture of $\mathbf{M}(\mathbf{z_\theta}, g)$ is specified in Fig 7. The feature vector $\mathbf{z_\theta}$ and the non-discrete egomotion $g = \boldsymbol{y}_i - \boldsymbol{y}_j$ are processed through separate neural network modules $\mathbf{M}^1$ and $\mathbf{M}^2$ before appending and processing through a final module $\mathbf{M}^3$. The specific architectures of these internal modules $\mathbf{M}^1, \mathbf{M}^2, \mathbf{M}^3$ are specified in Sec 4.

It is worth reiterating that the architectures of equivariance maps define the nature of the desired equivariance, and control the strength of the equivariance objective: broadly, we expect that the more complex the architecture, the weaker the equivariance regularization is. An equivariance map architecture that is very simple could lead to heavy regularization, but equally, complex architectures might backpropagate no regularizing gradients to the base learned features $\mathbf{z_\theta}$, since they might be able to represented overfitted equivariance maps even for arbitrary input features, such as features from a randomly initialized neural network.

3.6 Applying learned equivariant representations to recognition tasks

While we have thus far described our formulation for generic equivariant image representation learning, our hypothesis is that representations trained as above will facilitate high-level visual tasks such as recognition. One way to see this is by observing that equivariant representations expose camera and object pose-related parameters to a recognition algorithm, which may then account for this critical information while making predictions. For instance, a feature space that embeds knowledge of how objects change under different viewpoints/manipulations may allow a recognition system to hallucinate new views (in that feature space) of an object to improve performance.

More generally, recall the intuitions gained from the view prediction task illustrated in Fig 2. As discussed in Sec 1, acquiring the ability to hallucinate future views in severely underdetermined situations requires mastery of complex visual skills like depth, 3D geometry, semantics, and context. Therefore, our equivariance formulation of this view prediction task within the learned feature space induces the development of these ancillary high-level skills, which are transferable to other high-level tasks like object or scene recognition.

Suppose that in addition to the ego-pose annotated pairs $\mathcal{U}$ we are also given a small set of N_l class-labeled static images, $\mathcal{L} = \{(\boldsymbol{x}_k, c_k\}_{k=1}^{N_l}$, where $c_k \in \{1, \ldots, C\}$. We may now adapt our equivariance formulation to enable the training of a recognition pipeline on $\mathcal{L}$. In our experiments, we do this in two settings, purely unsupervised feature extraction (Sec 4.4), and unsupervised *regularization* of the supervised recognition task (Sec 4.5). We now describe the approaches for these two settings in detail.

In both of the scenarios below, note that neither the supervised training data $\mathcal{L}$ nor the testing data for recognition are required to have any associated sensor data. Thus, our features are applicable to standard image recognition tasks.

3.6.1 Adapting unsupervised equivariant features for recognition

In the unsupervised setting, we first train representations by optimizing the equivariance objective of Eq (7) (or Eq (13) for the non-discrete case). We then directly represent the class-labeled images from $\mathcal{L}$ in our learned equivariant feature space. These features may then be input to a generic machine learning pipeline, such as a k-nearest neighbor classifier, that is to be trained for recognition using labeled data $\mathcal{L}$. Alternatively, the weights learned in the network may be finetuned using the labeled data $\mathcal{L}$, producing a neural network classifier.

This setting allows us to test if optimizing neural networks *only* for equivariant representations, with no explicit *discriminative* component in the loss function, still produces discriminative representations. Aside from testing the power of our equivariant feature learning objective in isolation, this setting allows a nice modularity between the feature learning step and category learning step. In particular, when learned prior to any recognition task, our features

can be used for easy "off-the shelf" testing of the unsupervised neural network directly as a feature extractor for new tasks. The user does not need to simultaneously optimize the embedding parameters and classifier parameters specific to his task. Moreoever, it requires no more computational resources than for the Siamese paired network scheme described in Sec 3.5 for learning equivariant representations.

3.6.2 Unsupervised equivariance regularization for recognition

Alternatively, we may *jointly* train representations for equivariance, as well as for discriminative ability geared towards a target recognition task. Let L_{e_d} denote the unsupervised equivariance loss of Eq (7). We can integrate our unsupervised feature learning scheme with the recognition task, by optimizing a misclassification loss together with L_{e_d}. Let W be a $C \times D$ matrix of classifier weights. We solve jointly for W and the maps $\mathcal{M}$:

$$(\boldsymbol{\theta}^*, W^*, \mathcal{M}^*) = \arg\min_{\boldsymbol{\theta}, W, \mathcal{M}} L_c(\boldsymbol{\theta}, W, \mathcal{L}) + \lambda L_e(\boldsymbol{\theta}, \mathcal{M}, \mathcal{U}), \tag{14}$$

where L_c denotes the softmax loss over the learned features:

$$L_c(W, \mathcal{L}) = -\frac{1}{N_l} \sum_{i=1}^{N_l} \log(\sigma_{c_k}(W \mathbf{z}_{\boldsymbol{\theta}}(\boldsymbol{x}_i)), \tag{15}$$

and $\sigma_{c_k}(.)$ is the softmax probability of the correct class.

$$\sigma_{c_i}(\boldsymbol{p_i}) = \exp(p_{c_i}) / \sum_{c=1}^{C} \exp(p_c). \tag{16}$$

The regularizer weight λ in Eq (14) is a hyperparameter.

In this setting, the unsupervised egomotion equivariance loss encodes a prior over the feature space that can improve performance on the supervised recognition task with limited training examples.

To optimize Eq (14), in addition to the Siamese net that minimizes L_e as above, the supervised softmax loss is minimized through a third replica of the $\mathbf{z}_{\boldsymbol{\theta}}$ layer stack with weights tied to the two Siamese networks stacks. Labelled images from $\mathcal{L}$ are fed into this stack, and its output is fed into a softmax layer whose other input is the class label. So while this is a more complete framework for applying our equivariant representations to recognition tasks, it is also more computationally intensive; it requires more memory, more computation per iteration, and more iterations for convergence due to the more complex objective function. The complete scheme is depicted in Fig 5.

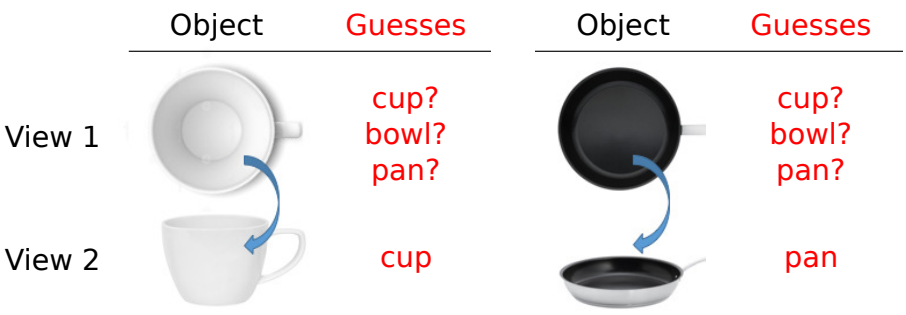

Fig. 8 Illustration of "next-best-view" selection for recognition. Suppose that a robot, having observed one view of an object (top) is not immediately confident of the category of the object. In the next-best view setting, it can then select to move around the object (or manipulate the object) intelligently, to disambiguate among its top competing hypotheses.

3.7 Equivariant representations for next-best view selection

Next, we model a situation where an agent equipped with equivariant visual representations has the ability to *act* on the real world at test time. Specifically, given one view of an object, the agent must decide how to move next to help recognize the object, i.e., which neighboring view would best reduce object category prediction uncertainty. This task is illustrated in Fig 8. Precisely because our features are equivariant (i.e., behave predictably) with respect to egomotion, we can exploit them to "envision" the next views that are possible and choose the most valuable one accordingly.

We now describe our method for this task, similar in spirit to [51]. We limit the choice of next view g to { "up", "down", "up+right" and "up+left" } for simplicity. First, we build a k-nearest neighbor (k-NN) image-pair classifier for each possible g, using only training image pairs $(\boldsymbol{x}, g\boldsymbol{x})$ related by the egomotion g. This classifier C_g takes as input a vector of length $2D$, formed by appending the features of the image pair (each image's representation is of length D) and produces the output probability of each class. So, $C_g([\mathbf{z}_{\boldsymbol{\theta}}(\boldsymbol{x}),\ \mathbf{z}_{\boldsymbol{\theta}}(g\boldsymbol{x})])$ returns class likelihood probabilities for all C classes. Output class probabilities for the k-NN classifier are computed from the histogram of class votes from the k nearest neighbors.

At test time, we first compute features $\mathbf{z}_{\boldsymbol{\theta}}(\boldsymbol{x}_0)$ on the given starting image $\boldsymbol{x}_0$. Next, we predict the feature $\mathbf{z}_{\boldsymbol{\theta}}(g\boldsymbol{x}_0)$ corresponding to each possible surrounding view g, as $M_g\mathbf{z}_{\boldsymbol{\theta}}(\boldsymbol{x}_0)$, per the definition of equivariance (cf. Eq (3)).

With these predicted transformed image features and the pair-wise nearest neighbor class probabilities $C_g(.)$, we may now pick the next-best view as:

$$g^* = \arg\min_g H(C_g([\mathbf{z}_{\boldsymbol{\theta}}(\boldsymbol{x}_0),\ M_g\mathbf{z}_{\boldsymbol{\theta}}(\boldsymbol{x_0})])), \tag{17}$$

where $H(.)$ is the information-theoretical entropy function. This selects the view that would produce the least predicted image pair class prediction uncertainty.

4 Experiments

We validate our approach on three public datasets and compare to multiple existing methods. The main questions we address in the experiments are: (i) quantitatively, how well is equivariance preserved in our learned embedding? (Sec 4.2); (ii) qualitatively, can we see the egomotion consistency of embedded image pairs? (Sec 4.3); (iii) when learned entirely without supervision, how useful are our method's features for recognition tasks? (Sec 4.4); (iv) when used as a regularizer for a classification loss, how effective are our method's features for recognition tasks? (Sec 4.5); and (v) how effective are the learned equivariant features for next-best view selection in an active recognition scenario? (Sec 4.6).

Throughout, we compare the following methods:

- CLSNET: A neural network trained only from the supervised samples with a softmax loss.
- TEMPORAL: The *temporal coherence* approach of [37], which regularizes the classification loss with Eq (9) setting the distance measure $d(.)$ to the ℓ_1 distance in d_1. This method aims to learn invariant features by exploiting the fact that adjacent video frames should not change too much.
- DRLIM: The approach of [17], which also regularizes the classification loss with Eq (9), but setting $d(.)$ to the ℓ_2 distance in d_1.
- LSM: The "learning to see by moving" (LSM) approach of [2], proposed independently and concurrently with our method, which also exploits video with accompanying egomotion for unsupervised representation learning. LSM uses egomotion in an alternative approach to ours; it trains a neural network to predict the observer egomotion g, given views $\boldsymbol{x}$ and $g\boldsymbol{x}$, before and after the motion. In our experiments, we use the publicly available KITTI-trained model provided by the authors.
- EQUIV: Our egomotion equivariant feature learning approach, as defined by the objective of Eq (7).
- EQUIV+DRLIM: Our approach augmented with temporal coherence regularization ([17]).
- EQUIV+DRLIM (non-discrete): The non-discrete motion variant of our approach as defined by the objective of Eq (12), augmented with temporal coherence regularization.[6]

All of these baselines are identically augmented with a classification loss for the regularization-based experiments in Sec 4.2, 4.5 and 4.6. TEMPORAL and DRLIM are the most pertinent baselines for validating our idea of exploiting egomotion for visual learning, because they, like us, use contrastive loss-based

[6] Note that we do not test EQUIV (non-discrete) i.e., the non-discrete formulation of Eq (13) in isolation. This is because Eq (13) specifies a non-contrastive loss that would result in collapsed feature spaces (such as $\mathbf{z}_{\boldsymbol{\theta}} = \mathbf{0} \forall \boldsymbol{x}$) if optimized in isolation. To overcome this deficiency, we optimize this non-discrete equivariance loss only in conjunction with the contrastive DRLIM loss in the EQUIV+DRLIM (non-discrete) approach.

formulations, but represent the popular "slowness"-based family of techniques ([55,5,16,33,12]) for unsupervised feature learning from video, which, unlike our approach, are passive. In addition, our results against LSM evaluate the strength of our egomotion-equivariance formulation against the alternative approach of [2].

4.1 Experimental setup details

Recall that in the fully unsupervised mode, our method trains with pairs of video frames annotated only by their ego-poses in $\mathcal{U}$. In the supervised mode, when applied to recognition, our method additionally has access to a set of class-labeled images in $\mathcal{L}$. Similarly, the baselines all receive a pool of unsupervised data and supervised data. We now detail the data composing these two sets.

Unsupervised datasets We consider two unsupervised datasets, NORB and KITTI, to compose the unlabeled video pools $\mathcal{U}$ augmented with egomotion.

- **NORB** [29]: This dataset has 24,300 96×96-pixel images of 25 toys captured by systematically varying camera pose. We generate a random 67%-33% train-validation split and use 2D ego-pose vectors $\boldsymbol{y}$ consisting of camera elevation and azimuth. Because this dataset has discrete ego-pose variations, we consider two egomotion patterns, i.e, $G = 2$ (cf. Sec 3.1): one step along elevation and one step along azimuth. For EQUIV, we use all available positive pairs for each of the two motion patterns from the training images, yielding a $N_u = 45,417$-pair training set. For DRLIM and TEMPORAL, we create a 50,000-pair training set (positives to negatives ratio 1:3). Pairs within one step (elevation and/or azimuth) are treated as "temporal neighbors", as in the turntable results of [17,37].
- **KITTI** [13,14]: This dataset contains videos with registered GPS/IMU sensor streams captured on a car driving around four types of areas (location classes): "campus", "city", "residential", "road". We generate a random 67%-33% train-validation split and use 2D ego-pose vectors consisting of "yaw" and "forward position" (integral over "forward velocity" sensor outputs) from the sensors. We discover egomotion patterns p_{ij} (cf. Sec 3.1) on frame pairs ≤ 1 second apart. We compute 6 clusters and automatically retain the $G = 3$ with the largest motions, which upon inspection correspond to "forward motion/zoom", "right turn", and "left turn" (see Fig 4). For EQUIV, we create a $N_u = 47,984$-pair training set with 11,996 positives. For DRLIM and TEMPORAL, we create a 98,460-pair training set with 24,615 "temporal neighbor" positives sampled $\leq$2 seconds apart.[7] Of the various KITTI cameras that simultaneously capture video, we use the feed from "camera 0" (see [14] for details) in our expriments. For our

[7] For fairness, the training frame pairs for each method are drawn from the same starting set of KITTI training videos.

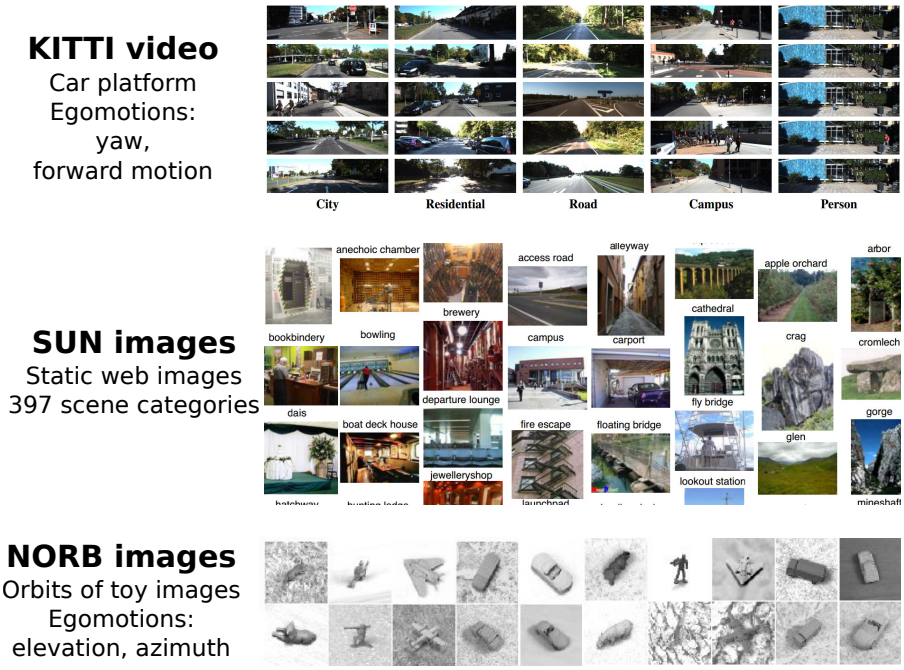

Fig. 9 We test our method across diverse standard datasets: (Top) Figure from [13] showcasing images from the four KITTI location classes (shown here in color; we use grayscale images), (Middle) Figure from [52] showcasing images from a subset of the 397 SUN classes (shown here in color; see text for image pre-processing details). (Bottom) Figure from [29], showing images of toys captured from varying camera poses and with varying backgrounds. In some of our experiments, we learn representations from KITTI videos and apply them to SUN scene recognition. Note how these two datasets vary greatly in content. In KITTI, the camera always faces a road, and it has a fixed field of view and camera pitch, and the content is entirely street scenes around Karlsruhe. In SUN, the images are downloaded from the internet, and belong to 397 diverse indoor and outdoor scene categories—most of which have nothing to do with roads.

unsupervised pretraining experiments, we reproduce the setting of [2] to allow fair comparison, cropping random 227×227 images from the original 370×1226 video frames to create the unlabeled dataset $\mathcal{U}$ ("KITTI-227"), before optimizing the objective of Eq (7). When testing the more computationally demanding regularization pipeline (cf. discussion in Sec 3.6) of 3.6.2 (Sec 4.5), we use grayscale, downsampled 32×32 pixel frames, so that (i) fast and thorough experiments are still possible, (ii) we can adopt CNN architecture choices known to be effective for tiny images [1], described below, and (iii) model complexity can be kept low enough so that our unsupervised training datasets are not too small.[8]

[8] Note that while the number of frame pairs N_u may be different for different methods, all methods have access to the same training videos, so this is a fair comparison. The differences in N_u are due to the methods themselves. For example, on KITTI data, EQUIV selectively

Supervised datasets In our recognition experiments, we consider three supervised datasets $\mathcal{L}$. These datasets allow us to test our approach's impact for three distinct recognition tasks for static images: object instance recognition, location recognition, and scene recognition. The supervised datasets are:

- **NORB**: We select six images from each of the $C = 25$ object training splits at random to create instance recognition training data.
- **KITTI**: We select four images from each of the $C = 4$ location class training splits at random to create location recognition training data.
- **SUN** [52]: We select six images for each of $C = 397$ scene categories at random from the standard training dataset to create scene recognition training data, unless otherwise stated. We preprocess them identically to the KITTI images above for all experiments. For the purely unsupervised experiments, we follow the setting of [2], resizing images to 256×256 before cropping random 227×227 regions ("SUN-227").

We keep all the supervised datasets small, since unsupervised feature learning should be most beneficial when labeled data is scarce. This corresponds to handling categorization problems in the "long tail". Note that while the video frames of the unsupervised datasets $\mathcal{U}$ are associated with ego-poses, the static images of $\mathcal{L}$ have no such auxiliary data.

Network architectures and optimization We now discuss the neural network architectures used for the base network $\mathbf{z}_{\boldsymbol{\theta}}$ and the equivariance maps in various experimental settings.

For NORB, $\mathbf{z}_{\boldsymbol{\theta}}$ is a fully connected network: 20 full-ReLU$\rightarrow$ D =100 full feature units. M_g is a single fully connected layer Linear(100,100). These are schematically depicted in Fig 10 (top row).

For KITTI, the base neural network $\mathbf{z}_{\boldsymbol{\theta}}$ closely follows the cuda-convnet [1] recommended CIFAR-10 architecture: 32 Conv(5x5)-MaxPool(3x3)-ReLU $\rightarrow$ 32 Conv(5x5)-ReLU-AvgPool(3x3) $\rightarrow$ 64 Conv(5x5)-ReLU-AvgPool(3x3) $\rightarrow$ D =64 full feature units. The equivariance map M_g is a single fully connected layer Linear(64,64), which takes in 64-dimensional $\mathbf{z}_{\boldsymbol{\theta}}(\boldsymbol{x})$ as input, and produces 64-dimensional $M_g\mathbf{z}_{\boldsymbol{\theta}}(\boldsymbol{x})$ as output. Fig 10 (middle row) presents schematics of these architectures.

For experiments with KITTI-227 and SUN-227, we follow the standard AlexNet architecture, augmented for fast training with batch normalization [20] (before every layer with learnable weights - *conv1-5*, *fc6*). We truncate the AlexNet architecture at the first fully connected layer, *fc6*, treating its output as the feature representation $\mathbf{z}_{\boldsymbol{\theta}}$. For EQUIV+DRLIM(discrete), the equivariance map modules M_g have the architecture: input $\rightarrow$ Linear(4096,128) $\rightarrow$ ReLU $\rightarrow$ Linear(128,4096), that produces a feature in the original 4096-dim feature space.[9] For EQUIV+DRLIM(non-discrete), the architecture of the equivariance map module $\mathbf{M}(.)$ follows the outline in Sec 3.5 and Fig 7. Specifically,

uses frame pairs corresponding to large motions (Sec 3.1), so even given the same starting videos, it is restricted to using a smaller number of frame pairs than DRLIM and TEMPORAL.

[9] We do not use a straightforward fully connected layer Linear(4096,4096) as this would drastically increase the number of network parameters, and possibly cause overfitting of M_g,

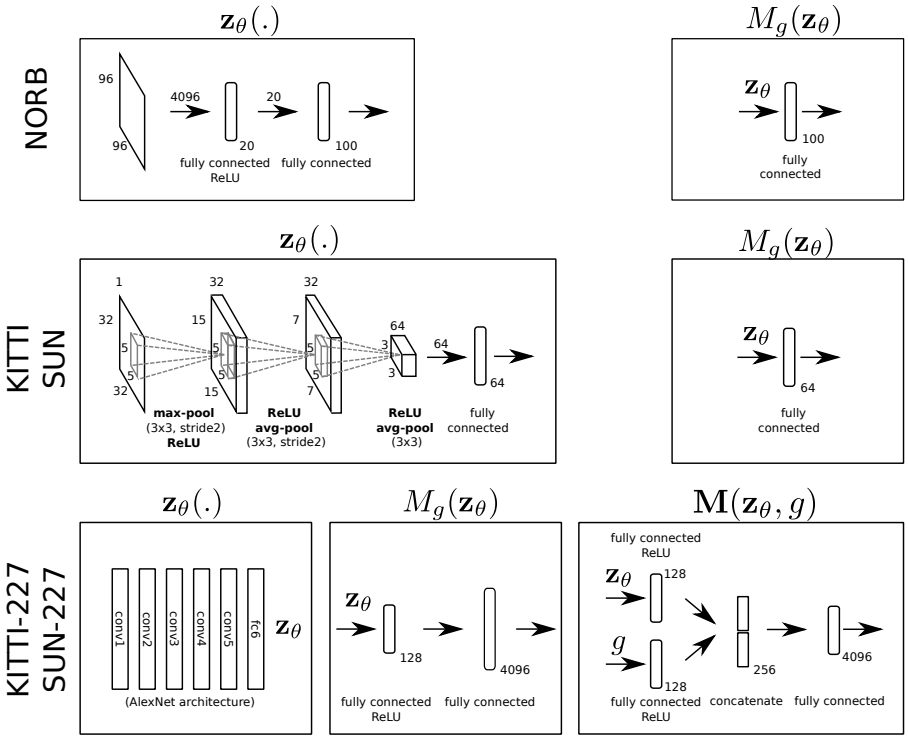

Fig. 10 Schematics representing neural network architectures used in various experimental settings, for the base network $\mathbf{z}_{\boldsymbol{\theta}}$ and the equivariance maps M_g (for the discrete case) and $\mathbf{M}(\mathbf{z}_{\boldsymbol{\theta}}, g)$ (for the non-discrete case).

- Module $\mathbf{M}^1$, which processes the 4096-dimensional output of *fc6*, has the architecture: input→ Linear(4096,128)→ReLU, producing a 128-dimensional output.
- Module $\mathbf{M}^2$, which processes the 2-dimensional continuous egomotion label, has the architecture: input→ Linear(2,128)→ReLU, producing a 128-dimensional output.
- Module $\mathbf{M}^3$, which processes the 256-dimensional concatenated outputs of the first two modules, has the architecture: input→ Linear(256,4096), producing a vector in the original 4096-dimensional feature space.

These architectures for KITTI-227 and SUN-227 experiments are shown in Fig 10 (bottom row).

We use Nesterov-accelerated stochastic gradient descent. The base learning rate and regularization λs are selected with greedy cross-validation.

We report all results for all methods based on five repetitions. For more details on architectures and optimization, see Appendix.

---

backpropagating poor equivariance regularization gradients through to the base network $\mathbf{z}_{\boldsymbol{\theta}}$. The M_g architecture we use in its place is non-linear due to the ReLU units.

4.2 Quantitative analysis: equivariance measurement

First, we test the learned features for equivariance. Equivariance is measured separately for each egomotion g through the normalized error ρ_g:

$$\rho_g = E\left[\frac{\|M_g^{'}\mathbf{z}_{\boldsymbol{\theta}}(\boldsymbol{x}) - \mathbf{z}_{\boldsymbol{\theta}}(g\boldsymbol{x})\|_2}{\|\mathbf{z}_{\boldsymbol{\theta}}(\boldsymbol{x}) - \mathbf{z}_{\boldsymbol{\theta}}(g\boldsymbol{x})\|_2}\right], \tag{18}$$

where $E[.]$ denotes the empirical mean, $M_g^{'}$ is the equivariance map, and $\rho_g = 0$ would signify perfect equivariance. To understand this error measure, we start by noting that the numerator is directly related to the definition of equivariance in Eq (3): $\mathbf{z}_{\boldsymbol{\theta}}(g\boldsymbol{x}) \approx M_g\mathbf{z}_{\boldsymbol{\theta}}(\boldsymbol{x})$. Thus, the numerator alone constitutes the most straightforward measure of equivariance error. However, this term depends on the *scale* of the feature representation, which may vary between methods. So, rather than measure the distance between the transformed and ground truth features directly, ρ_g measures the *ratio* by which M_g' *reduces* the distance between $\mathbf{z}_{\boldsymbol{\theta}}(g\boldsymbol{x})$ and $\mathbf{z}_{\boldsymbol{\theta}}(\boldsymbol{x})$. The denominator and numerator in Eq (18) are therefore the distance between the representations of original and transformed images, respectively *before* and *after* applying the equivariance map. The normalized error ρ_g is the empirical mean of this distance reduction ratio across all samples.

We closely follow the equivariance evaluation approach of [30] to solve for the equivariance maps of features produced by each compared method on held-out validation data, before computing ρ_g. Such maps are produced explicitly by our method, but not the baselines. Thus, as in [30], we compute their maps[10] by solving a least squares minimization problem based on the definition of equivariance in Eq (3):

$$M_g' = \arg\min_{M} \sum_{m(\boldsymbol{y}_i,\boldsymbol{y}_j)=g} \|\mathbf{z}_{\boldsymbol{\theta}}(\boldsymbol{x}_i) - M\mathbf{z}_{\boldsymbol{\theta}}(\boldsymbol{x}_j)\|_2. \tag{19}$$

The equivariance maps M_g' computed as above are used to compute the normalized errors ρ_g as in Eq (18). M_g' and ρ_g are computed on disjoint subsets of the validation image pairs.

We test both (i) "atomic" egomotions matching those provided in the training pairs (i.e, "up" 5°and "down" 20°) and (ii) composite egomotions ("up+right", "up+left", "down+right"). The latter lets us verify that our method's equivariance extends beyond those motion patterns used for training (cf. Sec 3.2).

First, as a sanity check, we quantify equivariance for the unsupervised loss of Eq (7) in isolation, i.e, learning with only $\mathcal{U}$. Our EQUIV method's average ρ_g error is 0.0304 and 0.0394 for atomic and composite egomotions in NORB, respectively. In comparison, DRLIM—which promotes invariance, not

[10] For uniformity, we do the same recovery of M_g' for our method; our results are similar either way.

Motion types →	atomic			composite			
Methods ↓	"up (u)"	"right (r)"	avg.	"u+r"	"u+l"	"d+r"	avg.
random	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
CLSNET	0.9276	0.9202	0.9239	0.9222	0.9138	0.9074	0.9145
TEMPORAL [37]	0.7140	0.8033	0.7587	0.8089	0.8061	0.8207	0.8119
DRLIM [17]	0.5770	0.7038	0.6404	0.7281	0.7182	0.7325	0.7263
EQUIV	0.5328	0.6836	**0.6082**	0.6913	0.6914	0.7120	**0.6982**
EQUIV+DRLIM	**0.5293**	**0.6335**	**0.5814**	**0.6450**	**0.6460**	**0.6565**	**0.6492**

Table 1 The "normalized error" equivariance measure ρ_g for individual egomotions (Eq (18)) on NORB, organized as "atomic" (motions in the EQUIV training set) and "composite" (novel) egomotions. This metric captures how well equivariance is preserved in the embedding space. Lower values are better.

equivariance—achieves $\rho_g = 0.3751$ and 0.4532. Thus, without class supervision, EQUIV tends to learn nearly completely equivariant features, even for novel composite transformations.

Next, we evaluate equivariance for all methods using features optimized for the NORB recognition task. Table 1 shows the results. As expected, we find that the features learned with EQUIV regularization are again easily the most equivariant. Normalized errors are lower for smaller motions than for larger motions, e.g., all methods do better on the atomic motion "u" (up by 5°) than on the other atomic motion "r" (right by 20°). Naturally, this also means error must be lower for atomic motions than for composite motions, since the latter are combinations of two atomic motions. This is confirmed by the results in Table 1.

Finally, we run similar experiments on the more challenging KITTI-227 data. Over the three egomotion clusters on KITTI-227, DRLIM *fc6* features achieved an average equivariance error ρ_g of 0.7791. In comparison, EQUIV produced significantly more equivariant features as expected, yielding average equivariance error 0.7315. To estimate how much more egomotion-equivariance may be beneficial for generic visual features, we now compare these unsupervised models against a fully supervised model ("IMAGENET-SUP" [27]) with the same standard AlexNet architecture as our models, but trained on ImageNet [9], a large manually curated classification dataset with millions of labeled images. Features extracted from such models are among the most widely used representations for various computer vision tasks today [10]. Fully supervised IMAGENET-SUP *fc6* features achieve 0.6285 average error, indicating significant egomotion-equivariance. We view the equivariance of these standard, widely used neural network features trained on labeled classification datasets as validation that that equivariance to egomotions may be a useful desideratum for learning good generic visual features in an unsupervised manner, as our method aims to do.

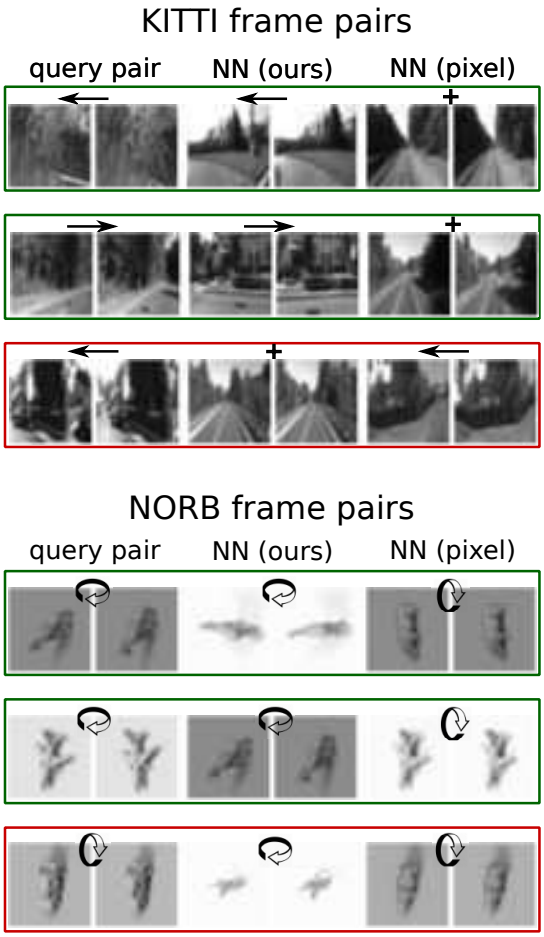

Fig. 11 Nearest neighbor image pairs (columns 3 and 4 in each block) in pairwise equivariant feature difference space for various query image pairs (columns 1 and 2 per block). For comparison, columns 5 and 6 show pixel-wise difference-based neighbor pairs. The direction of egomotion in query and neighbor pairs (inferred from ego-pose vector differences) is indicated above each block. See text.

4.3 Qualitative analysis: detecting image pairs with similar motions

To qualitatively evaluate the impact of equivariant feature learning, we pose a nearest neighbor task in the *feature difference* space to retrieve image pairs related by similar egomotion to a query image pair.

Given a learned feature space $\mathbf{z}(.)$ and a query image pair $(\boldsymbol{x}_i, \boldsymbol{x}_j)$, we form the pairwise feature difference $\boldsymbol{d}_{ij} = \mathbf{z}(\boldsymbol{x}_i) - \mathbf{z}(\boldsymbol{x}_j)$. In an equivariant feature space, other image pairs $(\boldsymbol{x}_k, \boldsymbol{x}_l)$ with similar feature difference vectors $\boldsymbol{d}_{kl} \approx$

$\boldsymbol{d}_{ij}$ would be likely to be related by similar egomotion to the query pair.[11] This can also be viewed as an analogy completion task, $\boldsymbol{x}_i : \boldsymbol{x}_j = \boldsymbol{x}_k :?$, where the right answer $\boldsymbol{x}_l$ must be computed by applying the unknown transformation $\boldsymbol{p}_{ij}$ to $\boldsymbol{x}_k$.

Fig 11 shows examples from KITTI (top) and NORB (bottom). For a variety of query pairs, we show the top neighbor pairs in the EQUIV space, as well as in pixel-difference space for comparison. Overall they visually confirm the desired equivariance property: neighbor-pairs in EQUIV's difference space exhibit a similar transformation (turning, zooming, etc.), whereas those in the original image space often do not. Consider the first NORB example (first row among NORB examples), where pixel distance, perhaps dominated by the lighting, identifies a wrong egomotion match, whereas our approach finds a correct match, despite the changed object identity, starting azimuth, lighting etc. The red boxes show failure cases. For instance, in the last KITTI example (third row), large foreground motion of a truck in the query image pair causes our method to wrongly miss the rotational motion.

4.4 Unsupervised feature extraction and finetuning for classification

Next, we present experiments to test whether useful visual features may be trained in neural networks by minimizing *only* the unsupervised equivariance loss of Eq (7), using *no* labeled samples. We follow the approach described in Sec 3.6.1.

As discussed before, this setting tests the power of our equivariant feature learning objective in isolation, and offers several advantages: (i) It has lower memory and computational requirements, since there is no need for a third stack of layers dedicated to classification (Sec 3.5, Fig 5). (ii) It allows easy off-the-shelf testing of the unsupervised neural network either directly as a feature extractor for new tasks, or as a "pretrained" network to be fine-tuned for new tasks (iii) It gets rid of the regularization λ of Eq (14), thus leaving fewer hyperparameters to optimize. These advantages allow relatively fast experimentation with large neural networks to test our purely supervised pipeline. We therefore perform experiments on large 227×227 images for most of the remainder of this section.

For these experiments, each layer stack follows the standard AlexNet architecture [27] for the layer stacks in these experiments, treating the output of the first fully connected layer, *fc6*, as the feature representation $\mathbf{z}_{\boldsymbol{\theta}}$ (as shown in Fig 10). Both the 227×227 image resolution and network architecture allow us to test our method with identical settings against a concurrent and independently proposed approach for unsupervised representation learning from video+egomotion [2], LSM. We compare the features produced by our method against baselines under two conditions: nearest neighbor classification, and

[11] Note that in our model of equivariance, this is not strictly true, since the pair-wise difference vector $M_g \mathbf{z}_{\boldsymbol{\theta}}(\boldsymbol{x}) - \mathbf{z}_{\boldsymbol{\theta}}(\boldsymbol{x})$ need not actually be consistent across images $\boldsymbol{x}$. However, for small motions and linear maps M_g, this still holds approximately, as we show empirically.

finetuning for classification. In the rest of this subsection, we first present both these settings in Sec 4.4.1 and Sec 4.4.2, before discussing their results together in Sec 4.4.3.

4.4.1 Nearest neighbor classifiers with unsupervised features

We first test our unsupervised features for the task of k-nearest neighbor scene recognition on SUN images ("SUN-227" as described in Sec 4.1). Nearest neighbor tasks are useful to directly analyze the effectiveness of the learned features; such tasks are also used in prior work for unsupervised feature learning [48,16]. Our nearest neighbor training set has 50 class-labeled training samples per class ($50 \times 397 = 19850$ total training samples), and we set $k = 1$. To evaluate the effect of the equivariance loss on features learned at various layers in the neural network, we perform these nearest neighbor experiments separately on features from *conv3*, *conv4*, *conv5*, and *fc6* layers of the AlexNet architecture used in our experiments.

In addition to the passive slow feature analysis baseline DRLIM, we also compare against the egomotion-based feature learning baseline, LSM, trained with identical settings to our method. We also report the performance when (i) using the pixel space itself as the feature vector ("pixel"), and (ii) using a randomly initialized neural network with identical architecture to ours and baselines ("random weights"). Note that this "random weights" baseline benefits from inductive biases specifically designed and encoded into the architecture of neural networks, such as through convolutions and pooling etc., same as our methods, which should enable it to produce better representations than its input pixel space ("pixel"), even without any training.

4.4.2 Finetuning unsupervised network weights for classification

In our second setting for testing the effectiveness of purely unsupervised training with our approach, we finetune the unsupervised network weights for a classification task.

Specifically, we build a new neural network classifier from the unsupervised network by attaching a small neural network "TopNet" with random weights to the layer that is to be evaluated. The architecture for TopNet is Linear(D,500)-ReLU-Linear(500,C)-Softmax Loss, where D is the dimensionality of the output at the layer under evaluation, and $C = 397$ is the number of classes in SUN. We finetune all models on 5 class-labeled training samples per class ($5 \times 397 = 1985$ total training samples). We used identical, standard finetuning settings for all models: learning rate 0.001 and momentum 0.9 with minibatch size 128 for 100 epochs with standard stochastic gradient descent. As before, we test all networks at various layers: *conv3*, *conv4*, *conv5*, and *fc6*. Once again, we compare our methods against DRLIM and LSM.

4.4.3 Unsupervised feature evaluation results

Datasets→	KITTI-227→SUN-227 [397 cls]				
Methods↓/Layers→	conv3	conv4	conv5	fc6	best layer
random	0.25				
pixels	1.80				
random weights	3.66 ± 0.12	3.60 ± 0.06	3.68 ± 0.25	4.04 ± 0.07	4.04 ± 0.07
LSM [2]	≈ 4.9	≈ 4.3	≈ 4.3	-	≈ 4.9
DRLIM [17]	6.44 ± 0.17	6.42 ± 0.23	5.80 ± 0.21	3.46 ± 0.10	6.44 ± 0.17
EQUIV	$\mathbf{7.14 \pm 0.22}$	$\mathbf{7.62 \pm 0.17}$	$\mathbf{6.96 \pm 0.09}$	$\mathbf{4.48 \pm 0.22}$	$\mathbf{7.62 \pm 0.17}$
EQUIV+DRLIM	$\mathbf{7.38 \pm 0.08}$	$\mathbf{7.48 \pm 0.19}$	$\mathbf{6.88 \pm 0.22}$	3.84 ± 0.18	$\mathbf{7.48 \pm 0.19}$
EQUIV+DRLIM (non-discrete)	6.46 ± 0.17	6.16 ± 0.09	6.22 ± 0.20	3.40 ± 0.08	6.46 ± 0.17

Table 2 SUN scene recognition accuracies with purely unsupervised feature learning, and nearest neighbor classification (k=1, 50 labeled training images per class) . The columns correspond to different layers of the AlexNet architecture. Our EQUIV-based methods once again outperform all baselines. (LSM *fc6* results are not reported in [2], so that entry is left blank. It has only one publicly shared model, so its scores do not have error bars.)

Datasets→	KITTI-227→SUN-227 [397 cls]				
Methods↓/Layers→	conv3	conv4	conv5	fc6	best layer
random	0.25				
LSM [2]	$\approx 0.26^*$	$\approx 1.04^*$	≈ 3.97	-	≈ 3.97
DRLIM [17]	6.16 ± 0.17	6.12 ± 0.08	5.61 ± 0.05	3.32 ± 0.05	6.16 ± 0.017
EQUIV	$\mathbf{6.77 \pm 0.05}$	$\mathbf{6.75 \pm 0.17}$	$\mathbf{6.77 \pm 0.07}$	$\mathbf{4.22 \pm 0.06}$	$\mathbf{6.77 \pm 0.05}$
EQUIV+DRLIM	$\mathbf{6.70 \pm 0.05}$	$\mathbf{6.71 \pm 0.08}$	6.36 ± 0.05	3.93 ± 0.06	$\mathbf{6.71 \pm 0.08}$
EQUIV+DRLIM (non-discrete)	6.10 ± 0.08	6.00 ± 0.03	5.37 ± 0.07	3.53 ± 0.06	6.10 ± 0.08

Table 3 SUN scene recognition accuracies with purely unsupervised feature learning, followed by finetuning for classification (5 labeled training images per class). The columns correspond to different layers of the AlexNet architecture. Our EQUIV-based methods once again outperform all baselines. (LSM *fc6* results are not reported in [2], so that entry is left blank. It has only one publicly shared model, so its scores do not have error bars.) * denotes models that failed to converge with finetuning.

Results for the nearest neighbor experiments in Sec 4.4.1 are shown in Table 2, and those for the finetuning experiments are in Table 3.

Our method strongly outperforms the baselines in both settings. On nearest neighbor experiments, EQUIV and EQUIV+DRLIM earn best scores of 7.62% and 7.48% compared to the best two baselines, DRLIM and LSM, which score 6.44% and 4.9% respectively. Finetuning experiments yield similar results (Table 3), with EQUIV and EQUIV+DRLIM yielding best results of 6.77% and 6.71%, with other methods far behind. These results establish that the equivariance formu-

lation more effectively exploits egomotion than the motion-regression approach of [2] for this data, and succeeds in learning generic image features.[12]

As expected, the pixel space baseline "pixels", and the randomly initialized neural network "random weights" perform significantly worse than all other methods on the nearest neighbor task in Table 2. The results also confirm the effect of the inductive bias of the network architecture; the "random weights" baseline builds stronger representations than its input pixel space.

Finally, we also test the non-discrete variant of our approach, EQUIV+DRLIM (non-discrete). Recall from Sec 3.4, that this approach has an important advantage over EQUIV+DRLIM: it does not discretize the space of motions, so it may be able to more effectively exploit egomotion information. This variant therefore allows us to evaluate whether this advantage is important empirically for this dataset.

As shown in Table 2 and Table 3, EQUIV+DRLIM (non-discrete) is once again stronger than the baselines. However, it falls short of our standard EQUIV+DRLIM variant. This may be due to (i) the non-linear equivariance map (cf. Sec 4.1) being too complex and thus weakening the constraint on the learned feature space (cf. the discussion in Sec 3.2), (ii) the formulation of Eq (13) lacking the *contrastive* property of the loss of Eq (7); the feature space is held up from collapsing only by the contrastive term within DRLIM (Eq (9)), and (iii) the difficulty of learning a single effective function $\mathbf{M}$ in Eq (13), to encode feature transformations corresponding to all possible motions. Given the success of the discretized motion variant of our approach, we focus on it for all subsequent experiments.

Interestingly, representations learned by all methods are most discriminative at *conv3* and *conv4*, and drop off at higher layers, especially *fc6*, where for nearest neighbor classification in Table 2, "random weights" performs better or on par with most methods. This suggests that model complexity of the networks may be too high in this setting (reproduced from [2]), given the relatively modest size of the KITTI dataset(≈20,500 video frames). Despite this, EQUIV features perform reasonably well even at *conv5* — in both Table 2 and Table 3, EQUIV conv5 features are better than all baseline features at any layer, suggesting that our egomotion-equivariance idea exploits unsupervised KITTI data more efficiently than the baselines. In particular, the temporal coherence objective of DRLIM appears to induce a loss of discriminativeness in feature layers close to *fc6*, the layer to which the loss is applied. DRLIM is best at *conv3*, and while EQUIV+DRLIM performs similarly to EQUIV at lower layers, its performance significantly drops at higher layers compared to EQUIV. This is in keeping with our intuitions outlined in Sec 1; the DRLIM slow feature analysis objective targets invariance, too much of which can lead to a loss of useful information for class discrimination. LSM too shows similar trends, falling off steadily in feature discriminativeness from *conv3* to *conv5*, as seen in Table 2.

[12] For fairness, Table 3 uses identical finetuning settings for all models (see Sec 4.4.2). Compared to its results in Table 3, the LSM baseline achieves higher scores on a related experiment in [2], possibly due to differences in finetuning settings and train-test splits.

	Datasets→ Methods↓	NORB-NORB [25 cls]	KITTI-KITTI [4 cls]	KITTI-SUN [397 cls, top-1]	KITTI-SUN [397 cls, top-10]
Regularized	TEMPORAL [37]	35.47 ± 0.51	45.12 ± 1.21	1.21 ± 0.14	8.24 ± 0.25
	DRLIM [17]	36.60 ± 0.41	47.04 ± 0.50	1.02 ± 0.12	6.78 ± 0.32
	EQUIV	38.48 ± 0.89	50.64 ± 0.88	1.31 ± 0.07	8.59 ± 0.16
	EQUIV+DRLIM	$\mathbf{40.78 \pm 0.60}$	$\mathbf{50.84 \pm 0.43}$	$\mathbf{1.58 \pm 0.17}$	$\mathbf{9.57 \pm 0.32}$
	random	4.00	25.00	0.25	2.52
	CLSNET	25.11 ± 0.72	41.81 ± 0.38	0.70 ± 0.12	6.10 ± 0.67
Unsupervised	TEMPORAL [37]	42.97 ± 0.62	47.39 ± 0.53	0.79 ± 0.01	-
	DRLIM [17]	42.83 ± 0.76	46.83 ± 0.45	0.86 ± 0.03	-
	EQUIV	42.18 ± 0.32	43.25 ± 1.00	0.56 ± 0.01	-
	EQUIV+DRLIM	$\mathbf{44.08 \pm 0.31}$	$\mathbf{49.59 \pm 0.66}$	$\mathbf{0.87 \pm 0.01}$	-

Table 4 Recognition result for three datasets (mean ± standard error) of accuracy % over five repetitions. The last two columns are both results from the KITTI-SUN task, only with different accuracy metrics (top-1 and top-10). Each unsupervised method is tested in two configurations: unsupervised regularization for supervised learning as in Sec 3.6.2 (top), and purely unsupervised feature learning as in Sec 3.6.1 followed by nearest neighbor classification (bottom) with the same labeled training set as the regularization methods (top).

4.5 Equivariance as a regularizer for recognition

Having assessed the effectiveness of training purely with our unsupervised objective in Sec 4.4, we now test the unsupervised regularization pipeline of Sec 3.6.2 on three recognition tasks: NORB-NORB, KITTI-KITTI, and KITTI-SUN. The first dataset in each pairing is unsupervised, and the second is supervised. To allow the usage of less complex neural network architectures,[13] reduce computational requirements and enable faster experimentation, all these datasets have smaller images relative to the KITTI-227 and SUN-227 datasets in Sec 4.4. NORB has 96×96 images while KITTI and SUN are composed of 32×32 images.

Table 4 (top, "Regularized") shows the results. On all three datasets, our method significantly improves classification accuracy, not just over the no-prior CLSNET baseline, but also over the closest previous unsupervised feature learning methods.[14]

All the unsupervised feature learning methods yield large gains over CLSNET on all three tasks. However, DRLIM and TEMPORAL are significantly weaker than the proposed method. Those methods are based on the "slow feature analysis" principle [50]—nearby frames must be close to one another in the learned feature space. We observe in practice (see Appendix) that temporally close frames are mapped close to each other after only a few training epochs. This points to a possible weakness in these methods—even with parameters

[13] We observed in Sec 4.4 that performance dropped at higher layers, indicating that AlexNet model complexity might be too high.

[14] To verify the CLSNET baseline is legitimate, we also ran a Tiny Image nearest neighbor baseline on SUN as in [52]. It achieves 0.61% accuracy (worse than CLSNET, which achieves 0.70%).

(temporal neighborhood size, regularization λ) cross-validated for recognition, the slowness prior is too weak to regularize feature learning effectively, since strengthening it causes loss of discriminative information. In contrast, our method requires *systematic* feature space responses to egomotions, and offers a stronger prior.

The most exciting result is KITTI-SUN (the two rightmost columns in Table 4). The KITTI data itself is vastly more challenging than NORB due to its noisy ego-poses from inertial sensors, dynamic scenes with moving traffic, depth variations, occlusions, and objects that enter and exit the scene. Furthermore, the fact we can transfer EQUIV features learned without class labels on KITTI (street scenes from Karlsruhe, road-facing camera with fixed pitch and field of view) to be useful for a supervised task on the very different domain of SUN ("in the wild" web images from 397 categories mostly unrelated to streets) indicates the generality of our approach. Note that our method was also validated with larger images in this same KITTI-SUN setting in Sec 4.4.

Our best recognition accuracy of 1.58% on SUN is achieved with only 6 labeled examples per class. It is ≈30% better than the nearest competing baseline TEMPORAL and over 6 times better than chance. We also compute "Top-10" accuracy (last column) for SUN. This corresponds to the likelihood of the true class being among the top-10 most likely classes according to the classifier. Top-10 accuracy trends closely follow the standard top-1 accuracy result, with EQUIV+DRLIM scoring 9.57% compared to 8.24% for the best baseline, TEMPORAL.

Comparison with purely unsupervised training: Finally, while the broad trends observed above are similar to those observed for the unsupervised training evaluation in Sec 4.4, individual accuracies are not directly comparable with those reported in Table 2 and 3, due to the differences in image sizes and network architectures. To address this, we now repeat the experiments of Sec 4.4.1 for these new datasets, performing nearest neighbor classification with purely unsupervised features. Networks are trained with purely unsupervised losses. Features are then extracted and used in a nearest neighbor classifier ($k = 1$) using the target task training set (6, 4, and 6 labeled training images per class respectively for NORB, KITTI, and SUN).

The results of these experiments, shown in Table 4 (bottom, "Unsupervised"), allow us to observe several trends. In general, we should expect that regularization yields higher accuracies than purely unsupervised feature learning followed by nearest neighbors — the regularization setting allows target domain and target task knowledge to influence the learning of the features themselves. From Table 4, KITTI-SUN and KITTI-KITTI both follow these expected trends. Of these, KITTI-SUN results in particular are consistently significantly poorer with purely unsupervised training, compared to regularization. We believe this is due to the large domain differences between KITTI and SUN, which mean that networks produced by purely unsupervised training on KITTI may be less well-suited to processing SUN image inputs. This is supported by the fact that accuracies on KITTI-KITTI, where only the target

dataset is changed and domains are matched, are only marginally better with regularization than with unsupervised training.

NORB-NORB accuracies are an exception in which purely unsupervised training consistently performs slightly better than regularization. We believe that the toy neural network architecture employed in our NORB experiments (see Sec 4.1 and Fig 10) could prevent effective training, allowing a simple nearest neighbor classifier to be more effective. Finally, on both the "regularized" and "unsupervised" accuracies in Table 4, EQUIV+DRLIM features improve significantly over EQUIV. This trend is thus consistent among experiments with small image datasets, but not observed in our experiments with larger images in Sec 4.4. This suggests that the performance of EQUIV+DRLIM may especially depend significantly on network architectures and/or feature dimensionality.

Varying training set sizes: We are especially interested in the impact of our feature learning idea when supervised training sets are relatively small. This corresponds to handling categorization problems in the "long tail", where training samples are scarce and priors are most useful. However, we do continue to see impact by our approach for larger training sets. For example, with N=20 samples for each of 397 classes on KITTI-SUN (7,940 total labeled training images), EQUIV scores 3.66+/-0.08% accuracy vs. 1.66+/-0.18 for CLSNET. Thus, our equivariance prior continues to boost recognition accuracy even at larger training set sizes.

4.6 Next-best view selection for recognition

Finally, we show the results of a direct application of equivariant features to "next-best view selection" on NORB, as described in Sec 3.7 and illustrated in Fig 8. Given one view of a NORB toy, the task is to tell a hypothetical robot how to move next, in order to best recognize the object, i.e, which neighboring view would best reduce object instance label prediction uncertainty.

To use the approach of Sec 3.7 for the baselines, we first compute equivariance maps M'_g for all methods as described in Sec 4.2. We set $k = 25$ for computing k-nearest neighbors, as per Sec 3.7.

Table 5 shows the results. On this task too, EQUIV features easily outperform the baselines. Recall that our approach for this task is based on exploiting the predictability of feature responses to observer motions, i.e., feature equivariance. This result thus highlights the potential for many such novel applications of our equivariant feature-learning formulation.

5 Conclusions and future work

Over the last decade, visual recognition methods have focused almost exclusively on learning from bags of images. We argue that such "disembodied"

Methods↓	1-view→ 2-view	accuracy gain
random	4.00 → 4.00	0
TEMPORAL [37]	29.60→ 31.90	2.30
DRLIM [17]	14.89→ 17.95	3.06
EQUIV	**38.52→43.86**	**5.34**
EQUIV+DRLIM	**38.46→43.18**	**4.72**

Table 5 Next-best view selection accuracy % on NORB. Our method EQUIV (and augmented with slowness in EQUIV+DRLIM) clearly outperforms all baselines.

image collections, though clearly valuable when collected at scale, deprive feature learning methods from the informative physical context of the original visual experience. We presented the first "embodied" approach to feature learning that generates features equivariant to egomotion. Our results on multiple datasets and on multiple tasks show that our approach successfully learns equivariant features, which are beneficial for many downstream tasks and hold great promise for future novel applications.

This work is only the first step towards embodied approaches for visual learning, and we hope to build further upon this in future work. One weakness of this work is the requirement of egomotion sensor streams registered with the video. This is completely free supervision, and such egomotion-registered video could in theory be captured on mobile devices, wearable cameras, and autonomous driving platforms. Yet, such data is currently not readily available in large quantities, thus limiting the capacity of models trained with our method. We have now begun to model equivariant feature learning with completely unlabeled web video [23], exploiting the fact that even though the specific camera motions may be unknown, a reasonable assumption is that those motions remain the same at nearby time instants in a video.

To see another research avenue that this work leaves open, recall the two advantages of the active kitten from Sec 1, which proved to be key to its perceptual development: (i) proprioceptive knowledge, and (ii) ability to *select* motions during development. While the current work models the first of these advantages by substituting known camera motion for the active kitten's proprioception, it currently relies on pre-recorded video that is captured without its direction. A particularly exciting research direction towards "embodied vision" that is now open to us, is to model an agent that not only *knows* how it is moving, but is also able to *select* its own motions to maximize visual learning. Note that this setting is distinct from the next-best view selection scenario presented above, in that the active choices would be made for the sake of learning, not solely for the sake of online decision making with a previously learned model. We have begun to explore this direction in [22], where an "active vision" agent is trained end-to-end to make intelligent motion choices to improve its own ability to classify objects and scenes. We plan to build further along these and other directions in future work.

Acknowledgements This research is supported in part by ONR PECASE Award N00014-15-1-2291. We also thank Texas Advanced Computing Center for their generous support, Pulkit Agrawal for sharing models and code and for helpful discussions, Ruohan Gao for helpful discussions, and our anonymous reviewers for their constructive suggestions.

References

1. Cuda-convnet. https://code.google.com/p/cuda-convnet/ 23, 24
2. Agrawal, P., Carreira, J., Malik, J.: Learning to see by moving. In: ICCV (2015) 7, 8, 21, 22, 23, 24, 29, 31, 32
3. Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L.D., Monfort, M., Muller, U., Zhang, J., et al.: End to end learning for self-driving cars. arXiv preprint arXiv:1604.07316 (2016) 7
4. Bromley, J., Bentz, J.W., Bottou, L., Guyon, I., LeCun, Y., Moore, C., Säckinger, E., Shah, R.: Signature verification using a Siamese time delay neural network. IJPRAI (1993) 16
5. Cadieu, C.F., Olshausen, B.A.: Learning intermediate-level representations of form and motion from natural movies. Neural computation (2012) 6, 11, 14, 22
6. Chen, C., Grauman, K.: Watching unlabeled videos helps learn new human actions from very few labeled snapshots. In: CVPR (2013) 6
7. Cohen, T.S., Welling, M.: Transformation Properties of Learned Visual Representations. ICLR (2015) 7
8. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR (2005) 5, 6
9. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on (2009) 27
10. Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., Darrell, T.: Decaf: A deep convolutional activation feature for generic visual recognition. In: ICML (2014) 27
11. Dosovitskiy, A., Springenberg, J.T., Riedmiller, M., Brox, T.: Discriminative Unsupervised Feature Learning with Convolutional Neural Networks. NIPS (2014) 6
12. Gao, R., Jayaraman, D., Grauman, K.: Object-centric representation learning from unlabeled videos. In: ACCV (2016) 6, 11, 14, 22
13. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets Robotics: The KITTI Dataset. IJRR (2013) 5, 9, 22, 23
14. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the KITTI vision benchmark suite. CVPR (2012) 5, 9, 22
15. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. AISTATS (2010) 40
16. Goroshin, R., Bruna, J., Tompson, J., Eigen, D., LeCun, Y.: Unsupervised Learning of Spatiotemporally Coherent Metrics. ICCV (2015) 5, 6, 11, 14, 22, 30
17. Hadsell, R., Chopra, S., LeCun, Y.: Dimensionality Reduction by Learning an Invariant Mapping. CVPR (2006) 6, 13, 15, 16, 21, 22, 27, 31, 33, 36
18. Held, R., Hein, A.: Movement-produced stimulation in the development of visually guided behavior. Journal of comparative and physiological psychology (1963) 1, 2
19. Hinton, G.E., Krizhevsky, A., Wang, S.D.: Transforming Auto-Encoders. ICANN (2011) 5, 6
20. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: ICML (2015) 24
21. Jayaraman, D., Grauman, K.: Learning image representations tied to egomotion. In: ICCV (2015) 7
22. Jayaraman, D., Grauman, K.: Look-ahead before you leap: end-to-end active recognition by forecasting the effect of motion (2016) 36
23. Jayaraman, D., Grauman, K.: Slow and steady feature analysis: higher order temporal coherence in video. In: CVPR (2016) 36

24. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: Convolutional architecture for fast feature embedding. arXiv (2014) 17, 40
25. Kivinen, J.J., Williams, C.K.: Transformation equivariant boltzmann machines. ICANN (2011) 5, 6, 11
26. Kornhauser, C.C.A.S.A., Xiao, J.: Deepdriving: Learning affordance for direct perception in autonomous driving. ICCV (2015) 7
27. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: NIPS (2012) 27, 29
28. Kulkarni, T.D., Whitney, W., Kohli, P., Tenenbaum, J.B.: Deep convolutional inverse graphics network. NIPS (2015) 5, 6
29. LeCun, Y., Huang, F.J., Bottou, L.: Learning methods for generic object recognition with invariance to pose and lighting. CVPR (2004) 22, 23
30. Lenc, K., Vedaldi, A.: Understanding image representations by measuring their equivariance and equivalence. CVPR (2015) 5, 6, 11, 26
31. Levine, S., Finn, C., Darrell, T., Abbeel, P.: End-to-end training of deep visuomotor policies. arXiv preprint arXiv:1504.00702 (2015) 7
32. Li, Y., Fathi, A., Rehg, J.M.: Learning to predict gaze in egocentric video. In: ICCV (2013) 7
33. Lies, J.P., Häfner, R.M., Bethge, M.: Slowness and sparseness have diverging effects on complex cell learning. PLoS computational biology (2014) 6, 11, 14, 22
34. Lowe, D.: Object recognition from local scale-invariant features. In: ICCV (1999) 5, 6
35. Memisevic, R.: Learning to relate images. PAMI (2013) 7
36. Michalski, V., Memisevic, R., Konda, K.: Modeling Deep Temporal Dependencies with Recurrent Grammar Cells. NIPS (2014) 5, 7
37. Mobahi, H., Collobert, R., Weston, J.: Deep Learning from Temporal Coherence in Video. ICML (2009) 5, 6, 11, 14, 16, 21, 22, 27, 33, 36
38. Nakamura, T., Asada, M.: Motion sketch: Acquisition of visual motion guided behaviors. IJCAI (1995) 7
39. Ranzato, M., Szlam, A., Bruna, J., Mathieu, M., Collobert, R., Chopra, S.: Video (language) modeling: a baseline for generative models of natural videos. arXiv (2014) 5, 7
40. Ren, X., Gu, C.: Figure-Ground Segmentation Improves Handled Object Recognition in Egocentric Video. In: CVPR (2010) 7
41. Schmidt, U., Roth, S.: Learning rotation-aware features: From invariant priors to equivariant descriptors. CVPR (2012) 5, 6, 11
42. Simard, P., LeCun, Y., Denker, J., Victorri, B.: Transformation Invariance in Pattern Recognition - Tangent distance and Tangent propagation (1998) 6
43. Simard, P.Y., Steinkraus, D., Platt, J.C.: Best practices for convolutional neural networks applied to visual document analysis. ICDAR (2003) 6
44. Sohn, K., Lee, H.: Learning invariant representations with local transformations. ICML (2012) 6
45. Tulsiani, S., Carreira, J., Malik, J.: Pose induction for novel object categories. In: ICCV (2015) 6
46. Tuytelaars, T., Mikolajczyk, K.: Local invariant feature detectors: a survey. Foundations and trends in computer graphics and vision **3**(3), 177–280 (2008) 5, 6
47. Vincent, P., Larochelle, H., Bengio, Y., Manzagol, P.A.: Extracting and composing robust features with denoising autoencoders. ICML (2008) 6
48. Wang, X., Gupta, A.: Unsupervised learning of visual representations using videos. In: CVPR (2015) 5, 6, 11, 14, 30
49. Watter, M., Springenberg, J., Boedecker, J., Riedmiller, M.: Embed to control: A locally linear latent dynamics model for control from raw images. In: NIPS (2015) 7
50. Wiskott, L., Sejnowski, T.J.: Slow feature analysis: unsupervised learning of invariances. Neural computation (2002) 6, 33
51. Wu, Z., Song, S., Khosla, A., Tang, X., Xiao, J.: 3d shapenets for 2.5 d object recognition and next-best-view prediction. CVPR (2015) 20
52. Xiao, J., Hays, J., Ehinger, K.A., Oliva, A., Torralba, A.: Sun database: Large-scale scene recognition from abbey to zoo. CVPR (2010) 5, 23, 24, 33

53. Xu, C., Liu, J., Kuipers, B.: Moving object segmentation using motor signals. ECCV (2012) 7
54. Yamada, K., Sugano, Y., Okabe, T., Sato, Y., Sugimoto, A., Hiraki, K.: Attention prediction in egocentric video using motion and visual saliency. PSIVT (2012) 7
55. Zou, W., Zhu, S., Yu, K., Ng, A.Y.: Deep learning of invariant features via simulated fixations in video. NIPS (2012) 6, 11, 14, 22

Appendix

Optimization and hyperparameter selection

(Elaborating on para titled "Network architectures and Optimization" in Sec 4.1)

We use Nesterov-accelerated stochastic gradient descent as implemented in Caffe [24], starting from weights randomly initialized according to [15]. The base learning rate and regularization λs are selected with greedy cross-validation. Specifically, for each task, the optimal base learning rate (from 0.1, 0.01, 0.001, 0.0001) was identified for CLSNET. Next, with this base learning rate fixed, the optimal regularizer weight (for DRLIM, TEMPORAL and EQUIV) was selected from a logarithmic grid (steps of $10^{0.5}$). For EQUIV+DRLIM, the DRLIM loss regularizer weight fixed for DRLIM was retained, and only the EQUIV loss weight was cross-validated. The contrastive loss margin parameter δ in Eq (8) in DRLIM, TEMPORAL and EQUIV were set uniformly to 1.0. Since no other part of these objectives (including the softmax classification loss) depends on the scale of features,[15] different choices of margins δ in these methods lead to objective functions with equivalent optima - the features are only scaled by a factor. For EQUIV+DRLIM, we set the DRLIM and EQUIV margins respectively to 1.0 and 0.1 to reflect the fact that the equivariance maps M_g of Eq (7) applied to the representation $\mathbf{z}_{\boldsymbol{\theta}}(g\boldsymbol{x})$ of the transformed image must bring it closer to the original image representation $\mathbf{z}_{\boldsymbol{\theta}}(\boldsymbol{x})$ than it was before i.e., $\|M_g\mathbf{z}_{\boldsymbol{\theta}}(g\boldsymbol{x}) - \mathbf{z}_{\boldsymbol{\theta}}(\boldsymbol{x})\|_2 < \|\mathbf{z}_{\boldsymbol{\theta}}(g\boldsymbol{x}) - \mathbf{z}_{\boldsymbol{\theta}}(\boldsymbol{x})\|_2$.

In addition, to allow fast and thorough experimentation, we set the number of training epochs for each method on each dataset based on a number of initial runs to assess the scale of time usually taken before the classification softmax loss on validation data began to rise i.e., overfitting began. All future runs for that method on that data were run to roughly match (to the nearest 5000) the number of epochs identified above. For most cases, this number was of the order of 50000. Batch sizes (for both the classification stack and the Siamese networks) were set to 16 (found to have no major difference from 4 or 64) for NORB-NORB and KITTI-KITTI, and to 128 (selected from 4, 16, 64, 128) for KITTI-SUN, where we found it necessary to increase batch size so that meaningful classification loss gradients were computed in each SGD iteration, and training loss began to fall, despite the large number (397) of classes.

On a single Tesla K-40 GPU machine, NORB-NORB training tasks took $\approx$15 minutes, KITTI-KITTI tasks took $\approx$30 minutes, and KITTI-SUN tasks took $\approx$2 hours. The purely unsupervised training runs with large "KITTI-227" images took up to 15 hours per run.

[15] Technically, the EQUIV objective in Eq (7) may benefit from setting different margins corresponding to the different egomotion patterns, but we overlook this in favor of scalability and fewer hyperparameters.

The weakness of the slow feature analysis prior

We now present evidence supporting our claim in the paper that the principle of slowness, which penalizes feature variation within small temporal windows, provides a prior that is rather weak. In every stochastic gradient descent (SGD) training iteration for the DRLIM and TEMPORAL networks, we also computed a "slowness" measure that is independent of feature scaling (unlike the DRLIM and TEMPORAL losses of Eq (9) themselves), to better understand the shortcomings of these methods.

Given training pairs $(\boldsymbol{x}_i, \boldsymbol{x}_j)$ annotated as neighbors or non-neighbors by $n_{ij} = \mathbb{1}(|t_i - t_j| \leq T)$ (cf. Eq (9) in the paper), we computed pairwise distances $\Delta_{ij} = d(\mathbf{z}_{\boldsymbol{\theta}(s)}(\boldsymbol{x}_i), \mathbf{z}_{\boldsymbol{\theta}(s)}(\boldsymbol{x}_j))$, where $\boldsymbol{\theta}(s)$ is the parameter vector at SGD training iteration s, and $d(.,.)$ is set to the ℓ_2 distance for DRLIM and to the ℓ_1 distance for TEMPORAL (cf. Sec 4).

We then measured how well these pairwise distances Δ_{ij} predict the temporal neighborhood annotation n_{ij}, by measuring the Area Under Receiver Operating Characteristic (AUROC) when varying a threshold on Δ_{ij}.

These "slowness AUROC"s are plotted as a function of training iteration number in Fig 12, for DRLIM and COHERENCE networks trained on the KITTI-SUN task. Compared to the standard random AUROC value of 0.5, these slowness AUROCs tend to be near 0.9 already even before optimization begins, and reach peak AUROCs very close to 1.0 on both training and testing data within about 4000 iterations (batch size 128). This points to a possible weakness in these methods—even with parameters (temporal neighborhood size, regularization λ) cross-validated for recognition, the slowness prior is too weak to regularize feature learning effectively, since strengthening it causes loss of discriminative information. In contrast, our method requires *systematic* feature space responses to egomotions, and offers a stronger prior.

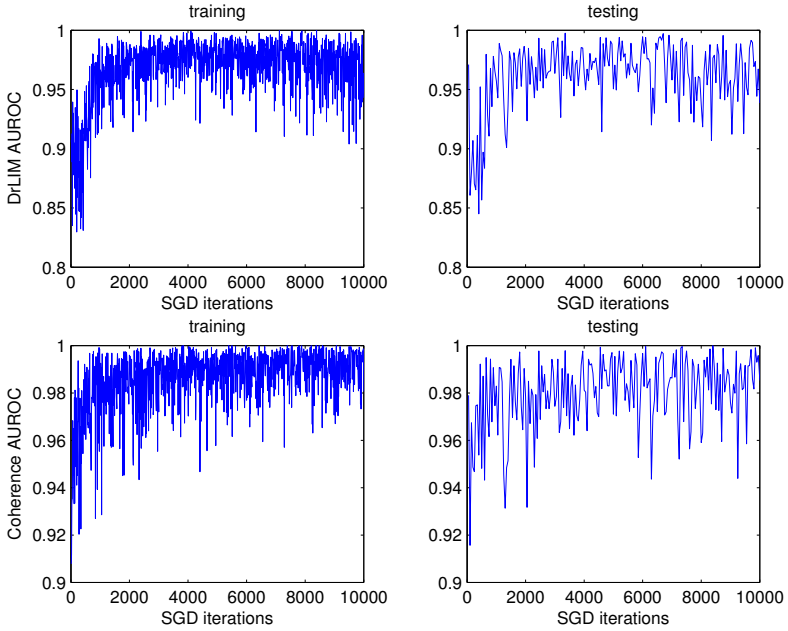

Fig. 12 Slowness AUROC on training (left) and testing (right) data for (top) DRLIM (bottom) COHERENCE, showing the weakness of slowness prior.