# CroCo v2: Improved Cross-view Completion Pre-training for Stereo Matching and Optical Flow

Philippe Weinzaepfel Thomas Lucas Vincent Leroy
Yohann Cabon Vaibhav Arora Romain Brégier Gabriela Csurka
Leonid Antsfeld Boris Chidlovskii Jérôme Revaud
NAVER LABS Europe
https://github.com/naver/croco

## Abstract

*Despite impressive performance for high-level downstream tasks, self-supervised pre-training methods have not yet fully delivered on dense geometric vision tasks such as stereo matching or optical flow. The application of self-supervised concepts, such as instance discrimination or masked image modeling, to geometric tasks is an active area of research. In this work, we build on the recent cross-view completion framework, a variation of masked image modeling that leverages a second view from the same scene which makes it well suited for binocular downstream tasks. The applicability of this concept has so far been limited in at least two ways: (a) by the difficulty of collecting real-world image pairs – in practice only synthetic data have been used – and (b) by the lack of generalization of vanilla transformers to dense downstream tasks for which relative position is more meaningful than absolute position. We explore three avenues of improvement. First, we introduce a method to collect suitable real-world image pairs at large scale. Second, we experiment with relative positional embeddings and show that they enable vision transformers to perform substantially better. Third, we scale up vision transformer based cross-completion architectures, which is made possible by the use of large amounts of data. With these improvements, we show for the first time that state-of-the-art results on stereo matching and optical flow can be reached without using any classical task-specific techniques like correlation volume, iterative estimation, image warping or multi-scale reasoning, thus paving the way towards universal vision models.*

## 1. Introduction

Self-supervised pre-training methods aim at learning rich representations from large amounts of unannotated data, which can then be finetuned on a variety of downstream tasks. This requires the design of pretext tasks, for

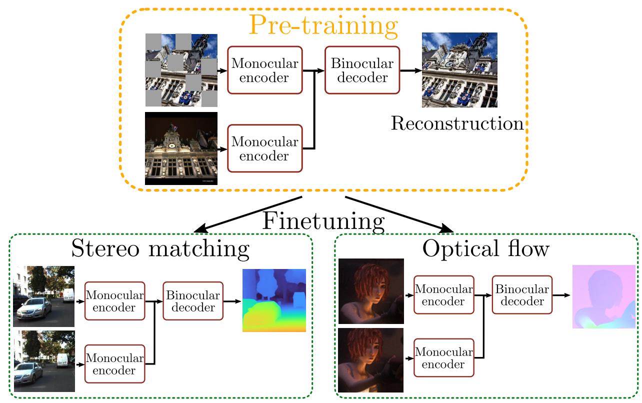


Figure 1. **Pre-training for dense geometric tasks.** We pre-train a generic architecture, with a monocular encoder and a binocular decoder, with cross-view completion before finetuning it on the stereo matching or optical flow downstream task.

which supervision signal can be extracted from the data itself, as well as generic architectures that can be easily transferred. We hypothesize that successfully pre-training large models for geometric tasks such as stereo matching or optical flow, see Figure 1, requires three things all together: (a) a well-designed dense pretext task inciting the understanding of 3D scene layout and geometry, (b) an architecture that processes pairs of images, suitable for different downstream tasks, and (c) large-scale real-world data.

Early self-supervised methods proceeded by discarding part of the signal (*e.g.* image color [95], patch ordering [55] or image orientation [25]) and trying to recover it. Later methods based on instance discrimination [12, 13, 16, 30] were first to surpass supervised pre-training on high-level tasks: they are based on the idea that output features should be invariant to well-designed classes of augmentations. Another recently successful pretext task is masked image modeling (MIM) [2, 22, 29, 81, 84, 100], where part of the input data is masked and an auto-encoder is trained to restore the full signal from the remaining visible parts. Instance discrimination and MIM methods have achieved excellent performance on semantic tasks such as image classification, in particular with limited amounts of annotated data [2, 17, 69], but have not led to breakthroughs in more geometric tasks like stereo matching and optical flow.

Adapting self-supervised pre-training to geometric vision tasks is an active area of research. Attempts have been made to design contrastive learning objectives at the pixel or patch level [80, 83, 85], but their performance gains have so far been more moderate than for global tasks. Besides, these gains are mainly demonstrated for dense *semantic* tasks such as semantic segmentation or object detection, rather than for geometric tasks such as depth estimation or stereo matching. Recently, [82] proposed the pretext task of cross-view completion (CroCo), a variant of MIM where a partially masked input image is reconstructed given visible patches and an additional view of the same scene. This pre-training objective is well suited to geometric downstream tasks as (a) it leverages pairs of images and (b) extracting relevant information from the second view requires geometric understanding of the scene. The CroCo architecture consists of a vision transformer (ViT) [20] encoder to extract features for the non-masked tokens of the first image, as well as for the second reference image, and a transformer to decode the features and reconstruct the masked image, as illustrated in Figure 2.

In spite of these advances, leveraging cross-view completion for geometric vision tasks remains challenging for at least two reasons. First, training with cross-view completion requires image pairs depicting the same scene; this can be hard to acquire at scale, yet scale is the cornerstone of the success of self-supervised pre-training. In practice, the CroCo model of [82] is pre-trained solely with synthetic data, which may limit its final performance. Second, most models trained with masking rely on ViTs [20], which typically use absolute positional embeddings. These do not generalize well to new image resolutions when finetuning, and are not always robust to cropping. This limits the applicability of current cross-view completion methods and may explain why the downstream tasks presented in [82] mostly use low-resolution squared images.

In this paper, we propose solutions to these limitations that enable to pre-train a large-scale cross-view completion model, see Figure 2, leading to state-of-the-art performance on stereo matching and optical flow. First, we tackle the problem of scalable pair collection, and gather millions of training pairs from different real-world datasets which cover various scenarios like indoor environments, street view data and landmarks, see Figure 3. To generate high-quality pre-training pairs, we carefully control the visual overlap for each pair of images. In fact, pairs with high overlap make the task trivial, whereas pairs with negligible overlap reduce it to standard MIM [82]. To measure this overlap, we leverage extra information available such as 3D meshes, additional sensors like LIDAR, or Structure-from-Motion (SfM)

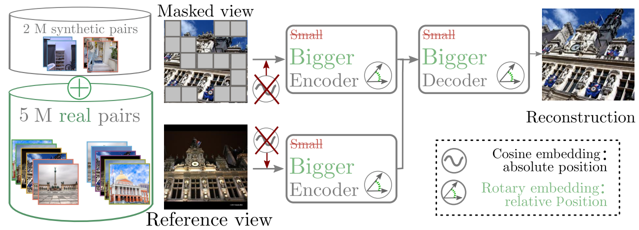


Figure 2. **Overview of the improvements in CroCo v2 for cross-view completion pre-training:** (a) collecting and using real-world images, (b) using rotary positional embeddings which model *relative* token positions, instead of absolute positions using the standard cosine embedding, (c) increasing network size both in the encoder and the decoder.

reconstructions for datasets with sufficient image coverage. From these data, we generate a set of high quality image pairs with sufficient overlap and viewpoint difference while also ensuring high diversity between pairs. Second, these large-scale datasets of pre-training pairs allow to scale up the model: (a) we use a larger encoder to extract better image features and (b) also scale up the decoder, which is responsible for combining information coming from the two views. Third, instead of the standard cosine positional embedding which encodes absolute positional information, we rely on the Rotary Positional Embedding (RoPE) [71] which efficiently injects relative positional information of token pairs in the attention mechanism.

We finetune our pre-trained model, referred to as CroCo v2, with this improved cross-view completion scheme on stereo matching and optical flow using a Dense Prediction Transformer (DPT) [59] head. Our models, termed CroCo-Stereo and CroCo-Flow, are simple and generic: we rely on a plain ViT encoder, followed by a plain transformer decoder which directly predicts the output (disparity for stereo, or optical flow) through the DPT head. We believe this is a meaningful step towards a universal vision model, *i.e.*, that can solve numerous vision tasks with a common architecture. In contrast to state-of-the-art methods for stereo matching or optical flow, our architecture does not rely on task-specific designs such as cost volumes [31, 34, 38, 39, 90], image warping [10, 74], iterative refinement [43, 46, 77] or multi-level feature pyramids [18, 43, 74]. While task-specific structures and prior knowledge may yield more data-efficient approaches, they come at the cost of being tailored to a single task. Our proposed pre-training allows us to eschew these and still reaches state-of-the-art performance on various stereo matching and optical flow benchmarks such as KITTI 2015 [53], ETH3D [65], Spring [52] or MPI-Sintel [11].

## 2. Related work

**Self-supervised learning.** The success of instance discrimination [12, 13, 16, 28, 30] has drawn a lot of atten-

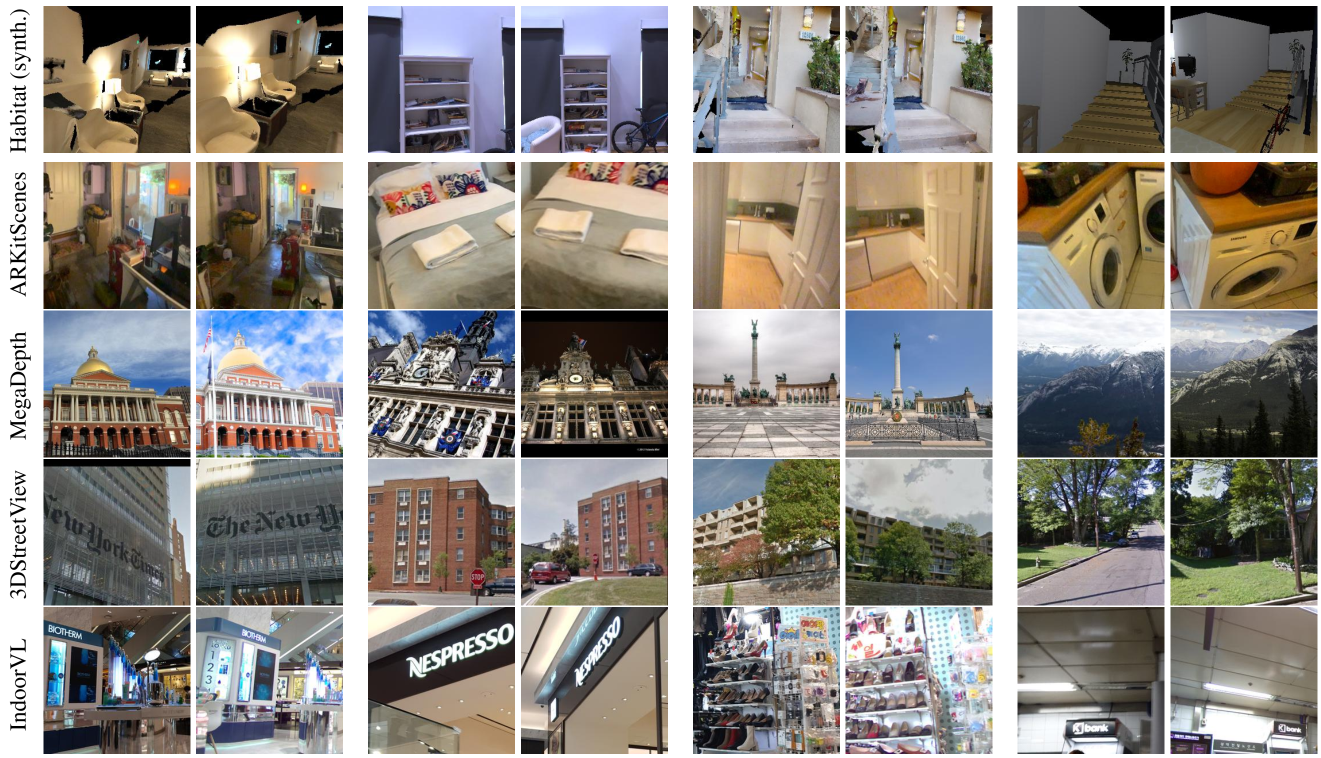


Figure 3. **Example of pre-training cropped image pairs** from Habitat which was the synthetic data used by CroCo [82] on the top row, and from real-world datasets we use in this paper (from ARKitScenes, MegaDepth, 3DStreetView and IndoorVL) below.

tion to self-supervised learning in computer vision [35]. In that paradigm, variants of an image are obtained by applying different data augmentations. Features extracted from the different variants are trained to be similar, while being pushed away from features obtained from other images. Such self-supervised models are particularly well tailored to image-level tasks, such as image classification, and have led to state-of-the-art performance on various benchmarks. Recent studies suggest that this success could be due to the object-centric [56] and the balanced [1] nature of ImageNet [61] that is used for pre-training. Recently, inspired by BERT [19] in natural language processing, different masked modeling methods have been adapted to computer vision. MIM pre-training aims at reconstructing masked information from an input image either in the pixel space [3, 4, 15, 22, 29, 84], or in the feature space [2, 5, 81], and sometimes after quantization [7, 100]. Recent works combine this framework in a teacher-student approach [42, 44] with improved masking strategy [23, 36, 44]. Overall, MIM models perform well on classification tasks. They have obtained some success on denser tasks such as object detection [29] or human pose estimation [89], and have been applied to robotic vision [57] when pre-trained on related datasets. More recently, CroCo [82] introduces the pretext task of cross-view completion, where a second view of the same scene is added to MIM. This is well suited to geometric downstream tasks: to leverage the second view and improve reconstruction accuracy, the model has to implicitly be aware of the geometry of the scene. CroCo outperforms MIM pre-training on an array of geometric tasks. However, it relies on synthetic data only, which may be sub-optimal, and does not reach the performance of the best task-specific methods.

**Positional embeddings.** Since a ViT treats its input as an orderless set of image patches or tokens, positional embeddings are a necessary tool to keep track of the position of each patch token from the original image. They can be either learned [13, 20] or handcrafted, such as the cosine positional embeddings from the original transformer [78]. Both learned and cosine embeddings are added explicitly to the signal and contain absolute positional information. However, models for pixel-level dense computer vision tasks should be able to process various image resolutions and be robust to cropping. Thus, relative positional embeddings, *e.g.* [66], that consider distances between tokens are preferable. For instance, Bello *et al.* [9] achieve better object detection results using relative self-attention. Similarly, Swin Transformers [49] and Swin V2 [48] observed improved performance using relative positional embeddings, while [72] showed it to be crucial in the cross attention for optical flow. Recently, [71] introduced the Rotary Positional Embedding (RoPE): a transformation to each key and query features is applied according to their absolute position, in such a way that the pairwise similarity scores used in the attention computation only depend on the relative positions of the token pairs and on their feature similarity. RoPE thus models relative positions at any resolution.

**Stereo matching and optical flow** can both be seen as a dense correspondence matching problem [88]. However the priors about matching itself and the completion of unmatched regions differ. This explains why most models are dedicated to one specific task despite many similarities in the strategies [40, 93]. Dense matching is most often posed with correlation/cost volume estimation from which matches can be extracted [21, 51]. For stereo, this volume typically has three dimensions [34, 39, 90], the third dimension representing a discretization of the disparity level, or four dimensions [14, 38, 54]. For optical flow, each pixel of the first image can be associated to any pixel of the second, resulting in a 4D correlation volume. The complexity of building, storing and leveraging such volume motivated numerous methods revolving around the ideas of coarse-to-fine [6, 77, 96], warping [74], sparse formulation [33], random search [97], dimension separation [94], tokenization [31]. Interestingly, recent works [72, 87, 88] leverage cross-attention to facilitate inter-image information exchanges but still rely on a low-resolution correlation volume, followed by an iterative refinement similar to [77]. Unimatch [88] made an important step towards a unified architecture for flow and stereo, but still relies on task-dependent (a) cross-attention mechanisms, (b) correlation volume and (c) post-processing. We similarly use the same architecture for both tasks, but our standard transformer model without cost volume can be pre-trained with existing self-supervised approaches and directly finetuned as is.

Several works propose self-supervised methods for estimating depth using stereo pairs or videos [26, 27], stereo with matching priors [99], or optical flow [73, 47, 91] typically with an unsupervised reconstruction loss. The main difference between this paradigm and ours is that we aim to pre-train a task-agnostic model that can be finetuned to different tasks, while these approaches aim to remove supervision for a single task.

# 3. Cross-view completion pre-training at scale

Our proposed pre-training method is based on the recently introduced cross-view completion (CroCo) framework [82]. It extends MIM to pairs of images. Given two different images depicting a given scene, the two images are divided into sets of non-overlapping patches, denoted as tokens, and 90% of the tokens from the first image are masked. The remaining ones are fed to a ViT [20] encoder to extract features for the first image. Similarly, tokens from the second image are fed to the same encoder with shared weights, and a ViT decoder processes the two sets of features together to reconstruct the target. Figure 2 provides an overview of the pre-training stage. Compared to standard masked image modeling methods, this approach can leverage the information in the second view to resolve some of the ambiguities about the masked context. To leverage this information, the model has to implicitly reason about the scene geometry and the spatial relationship between the two views, which primes it well for geometric tasks.

**Training data.** Collecting pairs of images that are suitable for this approach is non-trivial. First, images have to be paired together without manual annotation; second, their visual overlap has to be carefully controlled for the pairs to be useful. In [82], only synthetic data generated with the Habitat simulator [62] is used, which restricts the variety of the pre-training data. In contrast, we propose an approach to this real-world image pairing problem, necessary to use cross-view completion at scale, as detailed in Section 3.1.

**Positional embeddings.** The architecture used in [82] adapts ViTs to process pairs of images, by using cross-attention inside the decoder. Following standard practices, in their work cosine positional embedding is added to the token features prior to the encoder and the decoder. This models absolute position while dense tasks must typically be robust to cropping or images of various resolutions. In Section 3.2, we describe how relative positional embeddings can be adapted to cross-view completion.

**Large-scale models.** Finally, we discuss scaling-up the model in Section 3.3. CroCo [82] uses a ViT-Base encoder (12 blocks, 768-dimensional features, 12 attention heads) and a decoder composed of 8 blocks with 512-dimensional features and 16 heads. Using our large-scale dataset of real-world image pairs, we are able to scale to larger ViT architectures and demonstrate consistent performance gain.

## 3.1. Collecting real-world image pairs

We now present our approach to automatically select image pairs from real-world datasets that are suitable for pre-training. To be useful, pairs need to depict the same scene with some partial overlap. The overlap should not be small to the point where the task boils down to auto-completion. It should not be high either to the point where the task becomes a trivial ‘copy and paste’, *i.e.*, without requiring any understanding of scene geometry. On top of that, diversity should be as high as possible among pairs. We propose to use datasets that offer ways of getting information about the geometry of the scene and the camera poses. This signal can be captured using additional sensors like LIDAR, or it can be extracted using structure-from-motion (SfM) techniques if the images offer enough coverage of the scene. We use this information to obtain an image pair quality score based on overlap and difference in viewpoint angle. We then use a greedy algorithm to select a diverse set of image pairs. Finally, we generate overlapping image crops by leveraging image matching. Figure 4 gives an overview of our approach and we detail each step below.

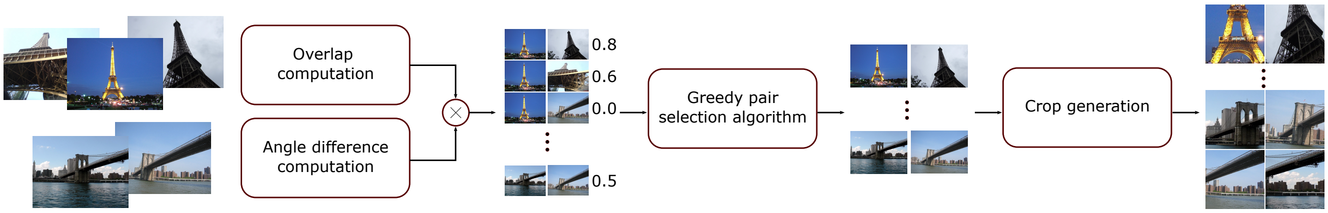


Figure 4. **Overview of our pre-training cropped image pair collection method.** Given a dataset of posed images, optionally with point clouds (*e.g.* from SfM) or meshes of the scene, we first measure the visual overlap between pairs and the viewpoint angle difference. Based on these scores, we use a greedy algorithm to select diverse image pairs and finally generate crops from them.

**Computing overlap scores.** The first step is to compute overlap scores for candidate pairs. We develop several approaches depending on the available information.

∘ *ARKitScenes* [8] provides 450,000 frames from 1,667 different indoor environments. The availability of the corresponding mesh for each frame enables the computation of the overlap between every pair of images. For each image $I$, we retrieve the set of mesh vertices $\mathcal{P}(I)$ that are visible. We then measure the intersection-over-union (IoU) of the vertices (3D points) for each pair of images $(I_1, I_2)$ as:

$$IoU(I_1, I_2) = \frac{|\mathcal{P}(I_1) \cap \mathcal{P}(I_2)|}{|\mathcal{P}(I_1) \cup \mathcal{P}(I_2)|}. \quad (1)$$

∘ *MegaDepth* [45] consists of around 300,000 images downloaded from the web corresponding to 200 different landmarks. From these images, a point cloud model for each landmark obtained using structure-form-motion (SfM) with COLMAP [64] is also provided. As above, it is possible to measure the vertex-based IoU between pairs of images, where each vertex is in this case a 3D point from the point cloud. Unfortunately, occlusions cannot be taken into account due to the absence of 3D mesh, which greatly degrades the overlap estimation. We propose a simple yet effective solution: we create an artificial occlusion model by attaching a ball of fixed radius to each 3D point, which occludes the vertices placed behind it. This way, we can compute a set of visible vertices for each image and evaluate the IoU as done previously.

∘ *3D Street View* [92] contains 25 million street view images from 8 cities. In addition to the camera pose, the 3D location and orientation (normal vector) of the target buildings are provided. To compute the overlap score, we create a pseudo 3D point cloud and apply the same technique as for MegaDepth. We start from an empty point cloud and append, for each target building, a $10 \times 6$ meters grid of $7 \times 11$ balls oriented according to the provided annotation.

∘ *Indoor Visual Localization datasets* (IndoorVL) [41] contains over 135,000 images from a large shopping mall and a large metro station in Seoul, South Korea, captured regularly with several months interval with 10 cameras and 2 laser scanners. The data is provided with accurate camera poses obtained via LiDAR SLAM refined by SfM-based optimization. We directly measure the overlap between images using the intersection between the camera frustrums using the accurate camera poses provided with the dataset. To encourage further diversity, we multiply this score by a factor 0.8 if both images come from the same capture session, thus favoring pairs taken with several months interval.

**Greedy image pair selection.** We rely on the overlap scores described above to select high quality pairs. This is however not sufficient: we also need pairs to be diverse, which would not be the case when randomly selecting good pairs, as images in the dataset can be very correlated. Therefore, we use a greedy algorithm to select non-redundant image pairs for pre-training. First, for each image pair $(I_1, I_2)$ we use a quality pair score $s$ given by:

$$s(I_1, I_2) = IoU(I_1, I_2) \times 4\cos(\alpha)\big(1 - \cos(\alpha)\big), \quad (2)$$

where $\alpha$ denotes the viewpoint angle difference between the two images (all the datasets above provide camera poses). The function $4\cos(x)(1-\cos(x))$ has a maximum value of 1 for $x = 60°$, 0 value for $x = 0°$ and $x = 90°$, and it is negative for angles above $90°$. This score thus favors pairs with different viewpoints while still having large overlaps. Given the score for every pair, we aim at building a large number of image pairs while ensuring diversity, *i.e.*, avoiding content redundancy. To do this, we use a greedy algorithm, where each time we select a pair of images with maximum score, we discard the two images forming the pair, as well as images that have too large IoU (above 0.75) with any of the two. We iteratively repeat this process until there is no pair with a score above a certain threshold.

**Crop generation per pair.** For pre-training, we use fixed-size crops of $224\times224$ pixels, as considering higher-resolution images would be too costly. In practice, we generate $256\times256$ crops and apply random cropping during pre-training. To generate crops on pairs of images while maintaining overlaps, we rely on quasi-dense keypoint matching, namely DeepMatching [60], except for pairs from ARKitScenes where we directly use matches from the mesh. Given the matches, we consider a grid of crops in the first image, estimate the corresponding matching crop in the second image and keep those with the most consistent matches and without overlap in the first image.

**Overall statistics.** In total, we collected about 5.3 million real-world pairs of crops with the process described above, with respectively 1,070,414 pairs from

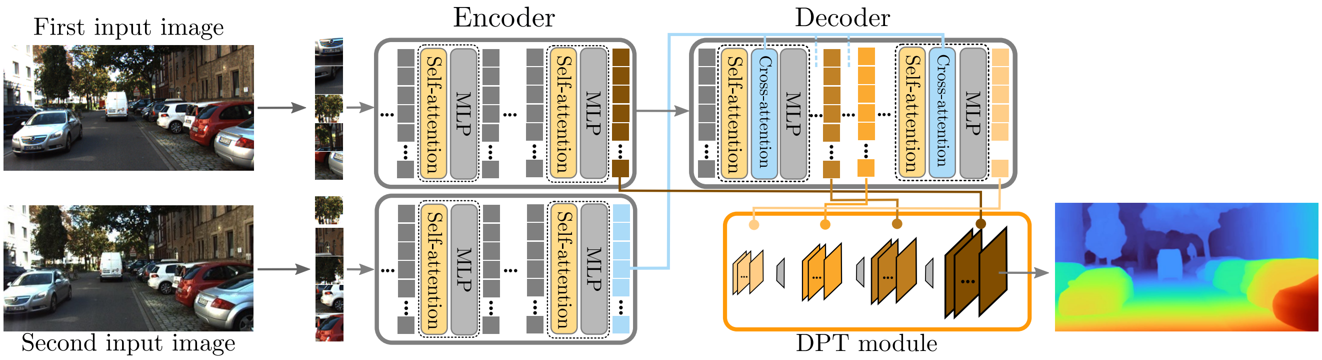


Figure 5. **Architecture of CroCo-Stereo and CroCo-Flow.** The two images (left and right views for stereo, two frames for flow) are split into patches and encoded with a series of transformer blocks with RoPE positional embeddings. The decoder consists in a series of transformer decoder blocks (self-attention among token features from the first image, cross-attention with the token features from the second image, and an MLP). Token features from different intermediate blocks are fed to the DPT module [59] to obtain the final prediction.

ARKitScenes [8], 2,014,789 pairs from MegaDepth [45], 655,464 from 3DStreetView [92], and 1,593,689 pairs from IndoorVL [41]. We added this to 1,821,391 synthetic pairs generated with the Habitat simulator [62], following the approach of [82]. Example pairs for each dataset are shown in Figure 3. They cover various scenarios, from indoor rooms – synthetic with Habitat or real with ARKitScenes – to larger crowded indoor environment (IndoorVL), landmarks (MegaDepth) and outdoor streets (3DStreetView).

### 3.2. Positional embeddings

We replace the cosine embeddings, which inject absolute positional information, by Rotary Positional Embedding (RoPE) [71]. RoPE efficiently injects information about the *relative* positioning of feature pairs when computing attention. Formally, let $q$ and $k$ represent a query and a key feature, at absolute positions $m$ and $n$ respectively. The main idea of RoPE is to design an efficient function $f(x, p)$ that transforms a feature $x$ according to its absolute position $p$ such that the similarity between the transformed query and the transformed key $\langle f(q, m), f(k, n)\rangle$ is a function of $q$, $k$ and $m - n$ only. [71] showed that a simple transformation such as applying rotations on pairs of dimensions according to a series of rotation matrices at different frequencies satisfy this desirable property. To deal with 2D signals such as images, we split the features into 2 parts, we apply the 1D positional embedding of the x-dimension on the first part, and the embedding of the y-dimension on the second part.

### 3.3. Scaling up the model

The combination of information extracted from the two images only occurs in the decoder. Following MAE [29], CroCo [82] uses a small decoder of 8 blocks consisting of self-attention, cross-attention and an MLP, with 512 dimensions and 16 attention heads. As the decoder is crucial for binocular tasks such as stereo or flow, we scale up the decoder and follow the ViT-Base hyper-parameters with 12 blocks, 768-dimensional features and 12 heads. We also scale up the image encoder from ViT-Base to ViT-Large, *i.e.*, increase the depth from 12 to 24, the feature dimension from 768 to 1024 and the number of heads from 12 to 16.

**Pre-training detailed setting.** We pre-train the network for 100 epochs with the AdamW optimizer [50], a weight decay of 0.05, a cosine learning rate schedule at a base learning rate of $3.10^{-4}$ with a linear warmup in the first 10 epochs, and a batch size of 512 spread on 8 GPUs. During pre-training, we simply use random crops and color jittering as data augmentation. We mask 90% of the tokens from the first image. Examples of cross-view completion obtained with our model are shown in the supplementary material.

## 4. Application to stereo matching and flow

We now present CroCo-Stereo and CroCo-Flow, our ViT-based correlation-free architecture for stereo matching and optical flow respectively, pre-trained with cross-view completion. This is much in contrast to current state-of-the-art methods which rely on task-specific design in the form of cost volumes [31, 34, 38, 39, 70, 74, 86, 90, 97], image warping [10, 74], iterative refinement [43, 46] and multi-level feature pyramids [18, 43, 74, 76]. Both CroCo-Stereo and CroCo-Flow share the same architecture.

**Architecture.** When finetuning the model for stereo or flow, both images are fed to the encoder as during pre-training (but without masking), and the decoder processes the tokens of both images. To output a pixel-wise prediction, we rely on DPT [59], which adapts the standard up-convolutions and fusions from multiple layers used in fully-convolutional approaches for dense tasks, to vision transformers. This allows to combine features from different blocks by reshaping them to different resolutions and fusing them with convolutional layers. In practice, we use the features from 4 blocks, regularly spread by an interval of a third of the decoder depth, starting from the last block, resulting in 1 block at the end of the encoder and 3 decoder blocks.

| pos. emb. | encoder | decoder | pre-train data | | Stereo (bad@1.0px↓) Md | ETH | SF(c) | SF(f) | Flow (EPE↓) FT(c) | FT(f) | Si.(c) | Si.(f) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| cosine | ViT-B | Small | 2M habitat | (CroCo [82]) | 26.3 | 1.82 | 6.7 | 7.0 | 3.89 | 3.56 | 2.07 | 2.57 |
| RoPE | ViT-B | Small | 2M habitat | | 25.3 | 0.60 | 6.0 | 6.3 | 3.73 | 3.37 | 2.13 | 2.77 |
| RoPE | ViT-B | Small | 2M habitat + 5.3M real | | 20.7 | 0.82 | 5.8 | 6.1 | 3.35 | 2.94 | 1.76 | 2.30 |
| RoPE | ViT-B | Base | 2M habitat + 5.3M real | | 17.1 | 1.14 | 5.3 | 5.6 | 3.10 | 2.73 | 1.51 | **1.99** |
| RoPE | ViT-L | Base | 2M habitat + 5.3M real | (**CroCo v2**) | **15.5** | **0.38** | **5.0** | **5.3** | **2.85** | **2.45** | **1.43** | **1.99** |

Table 1. **Ablative study** of each change to CroCo with the percentage of pixels with error above 1px (bad@1.0) on validation sets from Middlebury (Md), ETH3D, SceneFlow (SF) in clean (c) and final (f) renderings for stereo, and with the endpoint error (EPE) on validation sets from FlyingThings (FT) and MPI-Sintel (Si.) in both clean (c) and final (f) renderings for optical flow. A *Small* decoder has 8 decoder blocks with 16 attention heads on 512-dimensional features, while the *Base* one has 12 blocks with 12 heads on 768-dimensional features.

| nd < 400px | Bicyc2 ✓ | Compu ✓ | Austr ✓ | AustrP ✓ | Djemb ✓ | DjembL ✓ | Livgrm ✓ | Plants ✓ | Hoops | Stairs | Nkuba | Class | ClassE | Crusa | CrusaP | **avg↓** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| LEAStereo [18] | 1.83 | 3.81 | 2.81 | 2.52 | 1.07 | 1.64 | 2.59 | 5.13 | 5.34 | 2.79 | 3.09 | 2.46 | 2.75 | 2.91 | 3.09 | 2.89 |
| AdaStereo [70] | 2.19 | 2.29 | 4.37 | 3.08 | 1.40 | 1.64 | 3.93 | 7.58 | 4.46 | 2.67 | 3.69 | 3.29 | 3.35 | 3.78 | 2.94 | 3.39 |
| HITNet [76] | 1.43 | 1.87 | 3.61 | 3.27 | 0.90 | 9.12 | 2.37 | 4.07 | 4.45 | 3.38 | 3.45 | 2.43 | 3.20 | 4.67 | 4.74 | 3.29 |
| RAFT-Stereo [46] | 0.90 | 1.13 | 2.64 | 2.22 | **0.63** | 1.22 | 3.13 | 3.55 | 3.54 | 1.89 | 4.36 | **1.46** | 2.44 | 4.58 | 6.00 | 2.71 |
| CREStereo [43] | 1.38 | **1.06** | 2.63 | 2.53 | 0.64 | **1.11** | **1.42** | 5.31 | 3.22 | 2.40 | 2.51 | 1.92 | 2.31 | 1.78 | 1.83 | 2.10 |
| GMStereo [88] | 1.34 | 1.32 | 2.26 | 2.23 | 1.01 | 1.62 | 1.84 | 2.49 | **3.19** | 2.18 | 2.10 | 2.19 | **2.08** | **1.71** | **1.75** | **1.89** |
| **CroCo-Stereo** | **0.84** | 1.45 | **1.87** | **1.83** | 0.69 | 1.19 | 2.40 | **2.28** | 8.31 | **1.44** | **1.96** | 3.99 | 4.61 | 2.48 | 2.81 | 2.36 |

Table 2. **Evaluation on Middlebury** with the average error over all pixels for each sequence and the average (last column). Sequences are ordered according to their 'nd' value, which is the official threshold of maximum disparity used to clip predictions before evaluation.

**Loss.** We parameterize the output of the network with a Laplacian distribution [37]: given an input pair $(\boldsymbol{x}_1, \boldsymbol{x}_2)$, the model outputs a location parameter $\mu_i$ and a scale parameter $d_i$ per pixel location $i$ and is trained to minimize the negative log-likelihood of the ground-truth target disparity, denoted $\bar{\mu}$, under the predicted distribution:

$$-\log p(\bar{\mu}|\mu, d) = \sum_i \left[ \frac{|\mu_i - \bar{\mu}_i|}{d_i} - 2\log d_i \right]. \quad (3)$$

The scale parameter $d$ can be interpreted as an uncertainty score for the prediction: large errors are penalized less when $d$ is high, while good predictions are rewarded more if $d$ is low. It is thus optimal for the network to adapt the scale parameter. The second term comes from the normalization term of the Laplacian density and avoids the degenerate solution of always predicting a low scale parameter. Empirically, we find that using a probabilistic loss improves performance, see supplementary material for the ablation, and is useful for tiling strategies during inference, because it provides a per-pixel confidence estimate, as detailed below. A parameterization of $d_i$ ensures its positiveness: for stereo matching we use $d_i = e^{2\alpha(\sigma(d_i'/\alpha)-0.5)}$, with $\sigma$ the sigmoid function and $\alpha = 3$, and for optical flow $d_i = 1/\beta + (\beta - 1/\beta)\sigma(d_i')$ with $\beta = 4$, unless otherwise stated.

**Training.** We train CroCo-Stereo using 704×352 crops from various stereo datasets: CREStereo [43], SceneFlow [51], ETH3D [65], Booster [58], Middlebury (2005, 2006, 2014, 2021 and v3) [63]. We train CroCo-Flow using 384×320 crops from the TartanAir [79], MPI-Sintel [11], FlyingThings [51] and FlyingChairs [21] datasets. We refer to the supplementary material for more details on these datasets, the splits we use for the ablations, the data augmentation strategy, as well as training hyper-parameters.

**Inference.** We use a tiling-based approach. We sample overlapping tiles with the same size as the training crops in the first image. For each tile, we create a pair by sampling a corresponding tile at the same position from the second image. We then predict the disparity or flow between each pair of tiles. Such tiling approach was used *e.g.* in [31]. To merge the predictions done at a given pixel, we use a weighted average with weights $e^{-2\eta\alpha(\sigma(d_i'/\alpha)-0.5)}$ with $\eta = 5$ for stereo matching and $\alpha = 5, \eta = 2$ for optical flow, where $d_i'$ is the uncertainty predicted by the model.

## 5. Experiments

**Ablations.** We perform our ablations on the validation pairs (see the supplementary material for the splits we use) of Middlebury, ETH3D and SceneFlow for stereo matching, and FlyingThings and MPI-Sintel for optical flow. Table 1 reports the impact of the changes in CroCo v2 to improve CroCo [82] (pre-training data, positional embedding, larger encoder and decoder). We observe that they all lead to consistent improvements: replacing the cosine absolute positional embedding by RoPE, scaling up the decoder, using larger-scale pre-training data and a larger encoder. Altogether, this allows *e.g.* to improve performance as measured by the bad@1.0px metric from 26.3 to 15.5 on Middlebury (stereo matching), or the EPE from 2.07 to 1.43 on MPI-Sintel in its clean rendering (optical flow).

To further benchmark CroCo v2, we evaluate the pre-training of the encoder only on monocular tasks following the protocol of [4]. For semantic segmentation on

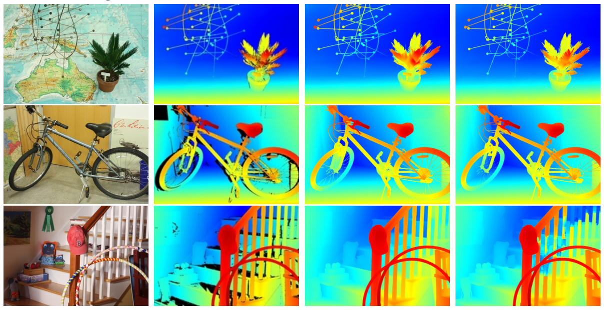


Figure 6. **Three example results from the Middlebury test set** (Australia, Bicycle2 and Hoops) with from left to right: the left image, the ground truth, CREStereo [43] and CroCo-Stereo.

| Method | D1-bg↓ | D1-fg↓ | D1-all↓ |
|---|---|---|---|
| AdaStereo [70] | 2.59 | 5.55 | 3.08 |
| HITNet [76] | 1.74 | 3.20 | 1.98 |
| PCWNet [67] | **1.37** | 3.16 | 1.67 |
| GMStereo [88] | 1.49 | 3.14 | 1.77 |
| ACVNet [86] | **1.37** | 3.07 | 1.65 |
| LEAStereo [18] | 1.40 | 2.91 | 1.65 |
| CREStreo [43] | 1.45 | 2.86 | 1.69 |
| **CroCo-Stereo** | 1.38 | **2.65** | **1.59** |

Table 3. **Evaluation on the KITTI 2015 stereo benchmark** with the percentage of outliers (*i.e.*, error above 3 pixels) for background (D1-bg), foreground (D1-fg) and all (D1-all) pixels.

ADE20k [98], we obtain 44.7 mean Intersection over Union *vs*. 40.6 for CroCo [82], and for monocular depth estimation on NYU v2 [68], we obtain 93.2 delta-1 *vs*. 90.1 for [82].

We provide in the supplementary material an ablation on the impact of pre-training (*i.e.*, a comparison with a randomly initialized network for finetuning), an ablation on the masking ratio during pre-training as well as a comparison between the L1 loss and Laplacian loss during finetuning.

**CroCo-Stereo *vs*. the state of the art.** We now evaluate CroCo-Stereo on the official leaderboards of Middlebury [63], KITTI 2015 [53], ETH3D [65] and Spring [52].

On Middlebury (Table 2), CroCo-Stereo obtains the lowest average error on 6 out of 15 sequences, in spite of using a generic patch-based transformer without any of the usual apparatus for stereo matching (*e.g*. cost-volume, coarse-to-scale processing, iterative refinement). However, in average, we obtain a worse error due to the fact that CroCo-Stereo produces really large errors for a few sequences like Hoops or ClassE. In fact, these errors correspond to cases with large maximum disparities (based on the maximum threshold value applied before evaluation), which is harmful for our simple tiling-based inference approach. This effect is visible in the prediction of the bottom example of Figure 6 where one can observe tiling artefacts, *e.g*. next to the stair pillars. In general, however, our method remains accurate, especially on thin structures like the pins on the map or the radius of the bicycle wheels in Figure 6.

| Method | bad@0.5 (%)↓ noc | bad@0.5 (%)↓ all | bad@1.0 (%)↓ noc | bad@1.0 (%)↓ all | avg err (px)↓ noc | avg err (px)↓ all |
|---|---|---|---|---|---|---|
| AdaStereo [70] | 10.22 | 10.85 | 3.09 | 3.34 | 0.24 | 0.25 |
| HITNet [76] | 7.89 | 8.41 | 2.79 | 3.11 | 0.20 | 0.22 |
| RAFT-Stereo [46] | 7.04 | 7.33 | 2.44 | 2.60 | 0.18 | 0.19 |
| DIP-Stereo [97] | 6.74 | 6.99 | 1.97 | 2.12 | 0.18 | 0.20 |
| GMStereo [88] | 5.94 | 6.44 | 1.83 | 2.07 | 0.19 | 0.21 |
| CREStereo [43] | 3.58 | 3.75 | **0.98** | **1.09** | **0.13** | **0.14** |
| **CroCo-Stereo** | **3.27** | **3.51** | 0.99 | 1.14 | 0.14 | 0.15 |

Table 4. **Evaluation on ETH3D** with the percentage of pixels with an error over 0.5px (bad@0.5), over 1px (bad@1.0) and the average error over non-occluded (noc) or all pixels.

| Method | 1px↓ | 1px s0-10↓ | 1px s10-40↓ | 1px s40+↓ | Abs↓ |
|---|---|---|---|---|---|
| RAFT-Stereo [46]‡ | 15.273 | 22.588 | 10.018 | 17.086 | 3.025 |
| AVC-Net [86]‡ | 14.772 | 18.386 | 11.346 | 18.145 | 1.516 |
| **CroCo-Stereo** | **7.135** | **2.934** | **7.757** | **13.247** | **0.471** |

Table 5. **Evaluation of CroCo-Stereo on the Spring benchmark** with the percentage of outliers (error over 1px) over all pixels, or over pixels with disparities in [0,10] (s0-10), in [10,40] (s10-40) and over 40 pixels (s40+), as well as the average absolute error (Abs). ‡ means methods submitted by the leaderboard's authors.

For KITTI 2015 (Table 3), we finetune CroCo-Stereo on $1216{\times}352$ crops from KITTI 2012 [24] and 2015 [53] for 20 epochs. CroCo-Stereo performs the best on the main D1-all metrics (outliers ratio at a 3px error threshold), with the best value also on foreground pixels, and at 0.01% of the best methods on background pixels.

For ETH3D, we use a Laplacian loss without bounds as it is limited to small disparities, *i.e*., with parameterization $d_i = e^{d'_i}$ and weights $e^{-3d'_i}$ for tiling. CroCo-Stereo sets a new state of the art for the ratio of pixels with an error over 0.5px (bad@0.5) and performs on par with CREStereo [43] for bad@1.0 and the average error, see Table 4. It outperforms recent approaches like GMStereo [88], RAFT-Stereo [46], DIP-Stereo [97] or HITNet [76] by a large margin, *e.g*. the bad@0.5 for non-occluded pixels is improved by 3% or more.

Finally, we report results on the recent Spring benchmark in Table 5 where our model is finetuned for 8 epochs on its training set. CroCo-Stereo outperforms the leading methods on all metrics with a large margin, *i.e*., the main bad@1 metric is reduced from 15% to 7% and the absolute error from 1.5 to 0.5px.

**CroCo-Flow *vs*. the state of the art.** We compare CroCo-Flow to the state of the art on the official leaderboards of MPI-Sintel [11], KITTI 2015 [53] and Spring [52].

On MPI-Sintel (Table 6), CroCo-Flow performs better than RAFT [77] which include many specialized refinement steps and use previous flow estimation as initialization. We rank second on the clean rendering and perform competitively on the final rendering, on par with most recent approaches such as GMFlow+ [88], SKFlow [75] or Flow-

| Method | clean↓ | final↓ |
|---|---|---|
| PWC-Net+ [74] | 3.45 | 4.60 |
| RAFT† [77] | 1.61 | 2.86 |
| CRAFT† [72] | 1.44 | 2.42 |
| FlowFormer [31] | 1.20 | **2.12** |
| SKFlow [75] | 1.30 | 2.26 |
| GMFlow+ [88] | **1.03** | **2.12** |
| **CroCo-Flow** | <u>1.09</u> | 2.44 |

Table 6. **Evaluation on the MPI-Sintel benchmark** with the EPE (↓) on the clean and final renderings. † means that the flow prediction from the previous frames is used as initialization.

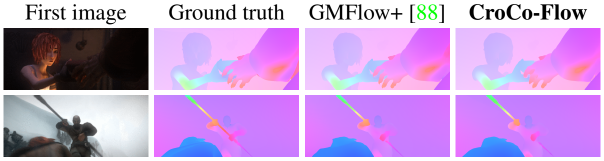


Figure 7. **Two examples from the MPI-Sintel test set** with from left to right: the first image, the ground truth, GMFlow+ [88] and CroCo-Flow.

| Method | Fl-bg↓ | Fl-fg↓ | Fl-all↓ |
|---|---|---|---|
| PWC-Net+ [74] | 7.69 | 7.88 | 7.72 |
| RAFT† [77] | 4.74 | 6.87 | 5.10 |
| CRAFT† [72] | 4.58 | <u>5.85</u> | 4.79 |
| FlowFormer [31] | 4.37 | 6.18 | 4.68 |
| GMFlow+ [88] | <u>4.27</u> | **5.60** | <u>4.49</u> |
| **CroCo-Flow** | **3.18** | 5.94 | **3.64** |

Table 7. **Evaluation of CroCo-Flow on the KITTI 2015 benchmark** with the percentage of outliers for background (F1-bg), foreground (F1-fg) and all (F1-all) pxiels. † means that the flow prediction from the previous frames is used as initialization.

| Method | 1px↓ | 1px s0-10↓ | 1px s10-40↓ | 1px s40+↓ | EPE↓ |
|---|---|---|---|---|---|
| FlowFormer [31]‡ | 6.510 | 3.381 | 5.530 | <u>35.344</u> | 0.723 |
| MS-Raft+ [32]‡ | <u>5.724</u> | <u>2.055</u> | <u>5.022</u> | 38.315 | <u>0.643</u> |
| **CroCo-Flow** | **4.565** | **1.225** | **4.332** | **33.134** | **0.498** |

Table 8. **Evaluation of CroCo-Flow on the Spring benchmark** with the number of outliers (error over 1px) over all pixels, or over pixels with flow norm in [0,10] (s0-10), in [10,40] (s10-40) and over 40 pixels (s40+) as well as the endpoint error (EPE). ‡ means methods submitted by the leaderboard's authors.

Former [31]. Figure 7 shows some visualizations of flow prediction.

For KITTI 2015 (Table 7), we finetuned the model on the training set from KITTI 2012 and 2015 for 150 epochs on crops of size 1216×352. CroCo-Flow performs best on the main F1-all metrics, *i.e.*, the percentage of outliers, with a large margin: the F1-all is reduced from 4.49% to 3.64% compared to GMFlow+ [88]. This gap mainly comes from the background pixels, while we perform on par with the best methods on foreground pixels.

Finally, on Spring, for which we finetune the model on its training set for 12 epochs, we obtain state-of-the-art performance, see Table 8. We obtain an EPE of 0.50, compared to 0.64 for the second best method, with an outlier ratio reduced for all flow norm ranges.

**Limitations.** The tiling-based inference strategy may prevent an accurate estimate in case of extremely large disparity or flow, where the corresponding pixels can be outside of the tile of the second image. A tiling strategy smarter than taking the same cropping coordinates in a pair of images could be considered.

## 6. Conclusion

For the first time, we have shown that large-scale pretraining can be successful for dense geometric tasks, thanks to a well-adapted pretext task and real-world data at scale. This enables to reach state-of-the-art performance with a ViT-based architecture without using task-specific designs, and thereby opening novel routes to tackle these problems, and new avenues towards more universal vision models.

## References


[1] Mahmoud Assran, Randall Balestriero, Quentin Duval, Florian Bordes, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, and Nicolas Ballas. The hidden uniform cluster prior in self-supervised learning. In *ICLR*, 2023. 3

[2] Mahmoud Assran, Mathilde Caron, Ishan Misra, Piotr Bojanowski, Florian Bordes, Pascal Vincent, Armand Joulin, Michael Rabbat, and Nicolas Ballas. Masked siamese networks for label-efficient learning. In *ECCV*, 2022. 1, 2, 3

[3] Sara Atito, Muhammad Awais, and Josef Kittler. SiT: Self-supervised vIsion Transformer. *arXiv preprint arXiv:2104.03602*, 2021. 3

[4] Roman Bachmann, David Mizrahi, Andrei Atanov, and Amir Zamir. MultiMAE: Multi-modal Multi-task Masked Autoencoders. In *ECCV*, 2022. 3, 7

[5] Alexei Baevski, Wei-Ning Hsu, Qiantong Xu, Arun Babu, Jiatao Gu, and Michael Auli. data2vec: A General Framework for Self-supervised Learning in Speech, Vision and Language. *arXiv preprint arXiv:2202.03555*, 2022. 3

[6] Shaojie Bai, Zhengyang Geng, Yash Savani, and J. Zico Kolter. Deep equilibrium optical flow estimation. In *CVPR*, 2022. 4

[7] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. BEiT: BERT Pre-Training of Image Transformers. In *ICLR*, 2022. 3

[8] Gilad Baruch, Zhuoyuan Chen, Afshin Dehghan, Tal Dimry, Yuri Feigin, Peter Fu, Thomas Gebauer, Brandon Joffe, Daniel Kurz, Arik Schwartz, et al. Arkitscenes–a diverse real-world dataset for 3d indoor scene understanding using mobile rgb-d data. In *NeurIPS*, 2021. 5, 6

[9] Irwan Bello, Barret Zoph, Ashish Vaswani, Jonathon Shlens, and Quoc V. Le. Attention Augmented Convolutional Networks. In *ICCV*, 2019. 3

[10] Thomas Brox, Andrés Bruhn, Nils Papenberg, and Joachim Weickert. High accuracy optical flow estimation based on a theory for warping. In *ECCV*, 2004. 2, 6

[11] Daniel J Butler, Jonas Wulff, Garrett B Stanley, and Michael J Black. A naturalistic open source movie for optical flow evaluation. In *ECCV*, 2012. 2, 7, 8

[12] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised Learning of Visual Features by Contrasting Cluster Assignments. In *NeurIPS*, 2020. 1, 2

[13] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging Properties in Self-Supervised Vision Transformers. In *ICCV*, 2021. 1, 2, 3

[14] J. Chang and Y. Chen. Pyramid stereo matching network. In *CVPR*, 2018. 4

[15] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative Pretraining From Pixels. In *ICML*, 2020. 3

[16] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020. 1, 2

[17] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E. Hinton. Big self-supervised models are strong semi-supervised learners. In *NeurIPS*, 2020. 2

[18] Xuelian Cheng, Yiran Zhong, Mehrtash Harandi, Yuchao Dai, Xiaojun Chang, Hongdong Li, Tom Drummond, and Zongyuan Ge. Hierarchical neural architecture search for deep stereo matching. In *NeurIPS*, 2020. 2, 6, 7, 8

[19] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL HLT*, 2019. 3

[20] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *ICLR*, 2021. 2, 3, 4

[21] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *ICCV*, 2015. 4, 7

[22] Alaaeldin El-Nouby, Gautier Izacard, Hugo Touvron, Ivan Laptev, Hervé Jegou, and Edouard Grave. Are Large-scale Datasets Necessary for Self-Supervised Pre-training? *arXiv preprint arXiv:2112.10740*, 2021. 1, 3

[23] Yuxin Fang, Li Dong, Hangbo Bao, Xinggang Wang, and Furu Wei. Corrupted Image Modeling for Self-Supervised Visual Pre-Training. In *ICLR*, 2023. 3

[24] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012. 8

[25] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised Representation Learning by Predicting Image Rotations. In *ICLR*, 2018. 1

[26] Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. In *CVPR*, 2017. 4

[27] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J Brostow. Digging into self-supervised monocular depth estimation. In *ICCV*, 2019. 4

[28] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Ávila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap Your Own Latent - A New Approach to Self-Supervised Learning. In *NeurIPS*, 2020. 2

[29] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked Autoencoders are Scalable Vision Learners. In *CVPR*, 2022. 1, 3, 6

[30] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum Contrast for Unsupervised Visual Representation Learning. In *CVPR*, 2020. 1, 2

[31] Zhaoyang Huang, Xiaoyu Shi, Chao Zhang, Qiang Wang, Ka Chun Cheung, Hongwei Qin, Jifeng Dai, and Hongsheng Li. FlowFormer: A transformer architecture for optical flow. In *ECCV*, 2022. 2, 4, 6, 7, 9

[32] Azin Jahedi, Maximilian Luz, Lukas Mehl, Marc Rivinius, and Andrés Bruhn. High resolution multi-scale raft (robust vision challenge 2022). *arXiv preprint arXiv:2210.16900*, 2022. 9

[33] Shihao Jiang, Yao Lu, Hongdong Li, and Richard Hartley. Learning optical flow from a few matches. In *CVPR*, 2021. 4

[34] Zequn Jie, Pengfei Wang, Yonggen Ling, Bo Zhao, Yunchao Wei, Jiashi Feng, and Wei Liu. Left-right comparative recurrent model for stereo matching. In *CVPR*, 2018. 2, 4, 6

[35] Longlong Jing and Yingli Tian. Self-supervised Visual Feature Learning with Deep Neural Networks: A Survey. *IEEE Trans. PAMI*, 2021. 3

[36] Ioannis Kakogeorgiou, Spyros Gidaris, Bill Psomas, Yannis Avrithis, Andrei Bursuc, Konstantinos Karantzalos, and Nikos Komodakis. What to Hide from Your Students: Attention-Guided Masked Image Modeling. In *ECCV*, 2022. 3

[37] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *CVPR*, 2018. 7

[38] A. Kendall, H. Martirosyan, S. Dasgupta, P. Henry, R. Kennedy, A. Bachrach, and A. Bry. End-to-end learning of geometry and context for deep stereo regression. In *ICCV*, 2017. 2, 4, 6

[39] Sameh Khamis, Sean Fanello, Christoph Rhemann, Adarsh Kowdle, Julien Valentin, and Shahram Izadi. Stereonet: Guided hierarchical refinement for real-time edge-aware depth prediction. In *ECCV*, 2018. 2, 4, 6

[40] Hamid Laga, Laurent Valentin Jospin, Farid Boussaid, and Mohammed Bennamoun. A survey on deep learning techniques for stereo-based depth estimation. *IEEE Trans. PAMI*, 2020. 4

[41] Donghwan Lee, Soohyun Ryu, Suyong Yeon, Yonghan Lee, Deokhwa Kim, Cheolho Han, Yohann Cabon, Philippe Weinzaepfel, Nicolas Guérin, Gabriela Csurka, et al. Large-scale localization datasets in crowded indoor spaces. In *CVPR*, 2021. 5, 6

[42] Youngwan Lee, Jeffrey Willette, Jonghee Kim, Juho Lee, and Sung Ju Hwang. Exploring the role of mean teachers in self-supervised masked auto-encoders. In *ICLR*, 2023. 3

[43] Jiankun Li, Peisen Wang, Pengfei Xiong, Tao Cai, Ziwei Yan, Lei Yang, Jiangyu Liu, Haoqiang Fan, and Shuaicheng Liu. Practical stereo matching via cascaded recurrent network with adaptive correlation. In *CVPR*, 2022. 2, 6, 7, 8

[44] Zhaowen Li, Zhiyang Chen, Fan Yang, Wei Li, Yousong Zhu, Chaoyang Zhao, Rui Deng, Liwei Wu, Rui Zhao, Ming Tang, and Jinqiao Wang. MST: Masked Self-Supervised Transformer for Visual Representation. In *NeurIPS*, 2021. 3

[45] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *CVPR*, 2018. 5, 6

[46] Lahav Lipson, Zachary Teed, and Jia Deng. Raft-stereo: Multilevel recurrent field transforms for stereo matching. In *3DV*, 2021. 2, 6, 7, 8

[47] Pengpeng Liu, Michael Lyu, Irwin King, and Jia Xu. Selflow: Self-supervised learning of optical flow. In *CVPR*, 2019. 4

[48] Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, et al. Swin transformer v2: Scaling up capacity and resolution. In *CVPR*, 2022. 3

[49] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021. 3

[50] Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. In *ICLR*, 2019. 6

[51] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *CVPR*, 2016. 4, 7

[52] Lukas Mehl, Jenny Schmalfuss, Azin Jahedi, Yaroslava Nalivayko, and Andrés Bruhn. Spring: A high-resolution high-detail dataset and benchmark for scene flow, optical flow and stereo. In *CVPR*, 2023. 2, 8

[53] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *CVPR*, 2015. 2, 8

[54] Guang-Yu Nie, Ming-Ming Cheng, Yun Liu, Zhengfa Liang, Deng-Ping Fan, Yue Liu, and Yongtian Wang. Multi-level context ultra-aggregation for stereo matching. In *CVPR*, 2019. 4

[55] Mehdi Noroozi and Paolo Favaro. Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzles. In *ECCV*, 2016. 1

[56] Senthil Purushwalkam and Abhinav Gupta. Demystifying contrastive self-supervised learning: Invariances, augmentations and dataset biases. In *NeurIPS*, 2020. 3

[57] Ilija Radosavovic, Tete Xiao, Stephen James, Pieter Abbeel, Jitendra Malik, and Trevor Darrell. Real-world robot learning with masked visual pre-training. *CoRL*, 2022. 3

[58] Pierluigi Zama Ramirez, Fabio Tosi, Matteo Poggi, Samuele Salti, Stefano Mattoccia, and Luigi Di Stefano. Open challenges in deep stereo: the booster dataset. In *CVPR*, 2022. 7

[59] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *ICCV*, 2021. 2, 6

[60] Jerome Revaud, Philippe Weinzaepfel, Zaid Harchaoui, and Cordelia Schmid. Deepmatching: Hierarchical deformable dense matching. *IJCV*, 2016. 5

[61] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 2015. 3

[62] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, Devi Parikh, and Dhruv Batra. Habitat: A Platform for Embodied AI Research. In *ICCV*, 2019. 4, 6

[63] Daniel Scharstein, Heiko Hirschmüller, York Kitajima, Greg Krathwohl, Nera Nešić, Xi Wang, and Porter Westling. High-resolution stereo datasets with subpixel-accurate ground truth. In *GCPR*, 2014. 7, 8

[64] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016. 5

[65] Thomas Schops, Johannes L Schonberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *CVPR*, 2017. 2, 7, 8

[66] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with Relative Position Representations. In *NAACL HLT*, 2018. 3

[67] Zhelun Shen, Yuchao Dai21, Xibin Song11, Zhibo Rao, Dingfu Zhou, and Liangjun Zhang. Pcw-net: Pyramid combination and warping cost volume for stereo matching. In *ECCV*, 2022. 8

[68] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from RGBD images. In *ECCV*, 2012. 8

[69] Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D. Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In *NeurIPS*, 2020. 2

[70] Xiao Song, Guorun Yang, Xinge Zhu, Hui Zhou, Zhe Wang, and Jianping Shi. Adastereo: a simple and efficient

approach for adaptive stereo matching. In *CVPR*, 2021. 6, 7, 8
[71] Jianlin Su, Yu Lu, Shengfeng Pan, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *arXiv preprint arXiv:2104.09864*, 2021. 2, 3, 6
[72] Xiuchao Sui, Shaohua Li, Xue Geng, Yan Wu, Xinxing Xu, Yong Liu, Rick Goh, and Hongyuan Zhu. Craft: Cross-attentional flow transformer for robust optical flow. In *CVPR*, 2022. 3, 4, 9
[73] Deqing Sun, Stefan Roth, and Michael J Black. A quantitative analysis of current practices in optical flow estimation and the principles behind them. *IJCV*, 2014. 4
[74] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Models matter, so does training: An empirical study of cnns for optical flow estimation. *IEEE Trans. PAMI*, 2019. 2, 4, 6, 9
[75] Shangkun Sun, Yuanqi Chen, Yu Zhu, Guodong Guo, and Ge Li. SKFlow: Learning optical flow with super kernels. In *NeurIPS*, 2022. 8, 9
[76] Vladimir Tankovich, Christian Hane, Yinda Zhang, Adarsh Kowdle, Sean Fanello, and Sofien Bouaziz. Hitnet: Hierarchical iterative tile refinement network for real-time stereo matching. In *CVPR*, 2021. 6, 7, 8
[77] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *ECCV*, 2020. 2, 4, 8, 9
[78] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. 2017. 3
[79] Wenshan Wang, Delong Zhu, Xiangwei Wang, Yaoyu Hu, Yuheng Qiu, Chen Wang, Yafei Hu, Ashish Kapoor, and Sebastian Scherer. Tartanair: A dataset to push the limits of visual slam. In *IROS*, 2020. 7
[80] Xinlong Wang, Rufeng Zhang, Chunhua Shen, Tao Kong, and Lei Li. Dense Contrastive Learning for Self-Supervised Visual Pre-Training. In *CVPR*, 2021. 2
[81] Chen Wei, Haoqi Fan, Saining Xie, Chao-Yuan Wu, Alan Yuille, and Christoph Feichtenhofer. Masked Feature Prediction for Self-Supervised Visual Pre-Training. In *CVPR*, 2022. 1, 3
[82] Weinzaepfel, Philippe and Leroy, Vincent and Lucas, Thomas and Brégier, Romain and Cabon, Yohann and Arora, Vaibhav and Antsfeld, Leonid and Chidlovskii, Boris and Csurka, Gabriela and Revaud Jérôme. CroCo: Self-Supervised Pre-training for 3D Vision Tasks by Cross-View Completion. In *NeurIPS*, 2022. 2, 3, 4, 6, 7, 8
[83] Zhenda Xie, Yutong Lin, Zheng Zhang, Yue Cao, Stephen Lin, and Han Hu. Propagate Yourself: Exploring Pixel-Level Consistency for Unsupervised Visual Representation Learning. In *CVPR*, 2021. 2
[84] Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. SimMIM: A Simple Framework for Masked Image Modeling. In *CVPR*, 2022. 1, 3
[85] Yuwen Xiong, Mengye Ren, and Raquel Urtasun. Loco: Local contrastive representation learning. In *NeurIPS*, 2020. 2
[86] Gangwei Xu, Junda Cheng, Peng Guo, and Xin Yang. Attention concatenation volume for accurate and efficient stereo matching. In *CVPR*, 2022. 6, 8
[87] Haofei Xu, Jing Zhang, Jianfei Cai, Hamid Rezatofighi, and Dacheng Tao. Gmflow: Learning optical flow via global matching. In *CVPR*, 2022. 4
[88] Haofei Xu, Jing Zhang, Jianfei Cai, Hamid Rezatofighi, Fisher Yu, Dacheng Tao, and Andreas Geiger. Unifying flow, stereo and depth estimation. *IEEE Trans. PAMI*, 2023. 4, 7, 8, 9
[89] Yufei Xu, Jing Zhang, Qiming Zhang, and Dacheng Tao. Vitpose: Simple vision transformer baselines for human pose estimation. In *NeurIPS*, 2022. 3
[90] Zhichao Yin, Trevor Darrell, and Fisher Yu. Hierarchical discrete distribution decomposition for match density estimation. In *CVPR*, 2019. 2, 4, 6
[91] Jason J Yu, Adam W Harley, and Konstantinos G Derpanis. Back to basics: Unsupervised learning of optical flow via brightness constancy and motion smoothness. In *ECCV workshop*, 2016. 4
[92] Amir R Zamir, Tilman Wekel, Pulkit Agrawal, Colin Wei, Jitendra Malik, and Silvio Savarese. Generic 3D representation via pose estimation and matching. In *ECCV*, 2016. 5, 6
[93] Mingliang Zhai, Xuezhi Xiang, Ning Lv, and Xiangdong Kong. Optical flow and scene flow estimation: A survey. *Pattern Recognition*, 2021. 4
[94] Feihu Zhang, Oliver J. Woodford, Victor Prisacariu, and Philip H. S. Torr. Separable flow: Learning motion cost volumes for optical flow estimation. In *ICCV*, 2021. 4
[95] Richard Zhang, Phillip Isola, and Alexei A. Efros. Colorful Image Colorization. In *ECCV*, 2016. 1
[96] Shiyu Zhao, Long Zhao, Zhixing Zhang, Enyu Zhou, and Dimitris N. Metaxas. Global matching with overlapping attention for optical flow estimation. In *CVPR*, 2022. 4
[97] Zihua Zheng, Ni Nie, Zhi Ling, Pengfei Xiong, Jiangyu Liu, Hao Wang, and Jiankun Li. DIP: deep inverse patchmatch for high-resolution optical flow. In *CVPR*, 2022. 4, 6, 8
[98] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ADE20K Dataset. In *CVPR*, 2017. 8
[99] Chao Zhou, Hong Zhang, Xiaoyong Shen, and Jiaya Jia. Unsupervised learning of stereo matching. In *ICCV*, 2017. 4
[100] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. iBOT: Image BERT Pre-training with Online Tokenizer. In *ICLR*, 2022. 1, 3