Federated Online Adaptation for Deep Stereo

Matteo Poggi[1,2] Fabio Tosi[1]
[1]Department of Computer Science and Engineering (DISI) [2]Advanced Research Center on Electronic System (ARCES)
University of Bologna, Italy
Project page: https://fedstereo.github.io/

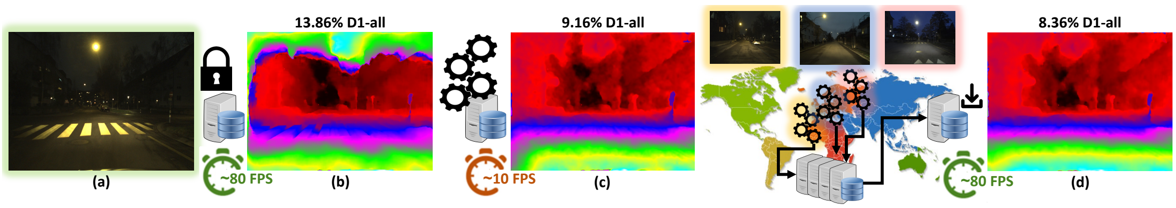

Figure 1. **Federated adaptation in challenging environments.** When facing a domain very different from those observed during training – e.g., nighttime images (a) – stereo models [55] suffer drops in accuracy (b). By enabling online adaptation [41] (c) the network can improve its predictions, at the expense of decimating the framerate. In our federated framework, the model can demand the adaptation process to the cloud, to enjoy its benefits while maintaining the original processing speed (d).

Abstract

We introduce a novel approach for adapting deep stereo networks in a collaborative manner. By building over principles of federated learning, we develop a distributed framework allowing for demanding the optimization process to a number of clients deployed in different environments. This makes it possible, for a deep stereo network running on resourced-constrained devices, to capitalize on the adaptation process carried out by other instances of the same architecture, and thus improve its accuracy in challenging environments even when it cannot carry out adaptation on its own. Experimental results show how federated adaptation performs equivalently to on-device adaptation, and even better when dealing with challenging environments.

1. Introduction

Depth sensing plays a key role in several applications in the fields of computer vision, robotics, and more. The use of stereo images [34] for this purpose has been one of the most studied topics for decades, consisting of matching pixels across two *rectified* images. This allows for estimating horizontal disparity between corresponding pixels and, consequently, their depth through triangulation. This process has been carried out through image processing algorithms [43] until nearly one decade ago, when deep learning started replacing hand-crafted solutions with neural networks [59]. The increasing growth of computational power in the hand of developers, together with the more and more annotated data becoming available, has rapidly established end-to-end deep networks [42] as the standard frameworks to deal with the problem [38, 44, 45].

In order to provide sufficient data for training deep stereo networks at their best, the use of synthetic datasets [35] has become a standard practice in the field. This, to some extent, also revealed one of the main limitations these models suffered from at first, which was the scarce capability to generalize to image domains very different from those observed at training time – a matter of concern common to other tasks involving deep networks, such as semantic segmentation [20]. First attempts to solve this shortcoming involved unsupervised adaptation techniques, either to be carried out offline [53, 56] or directly during deployment in real-time [41, 54, 55], with some computational overhead.

More recently, the community focused on dealing with the problem at its source – i.e., during the training process itself, by designing specific strategies to drive the deep network learning domain-invariant features [5, 11, 31, 69, 70] while, eventually, the most modern stereo networks [24, 30, 63, 65] can generalize much better than their predecessors. Despite these advancements, in the presence of very challenging conditions never observed during training, such as low illumination, sensor noise occurring at night, or the reflections appearing on rainy roads, we argue generalization capability alone might be insufficient. In such cases, online adaptation could still play a role, although at the cost of dropping the framerate at which the deep network operates. This price to pay might be reduced by means of specific adaptation strategies [41, 55] and allow for maintaining real-time processing when high-end GPUs are available, yet might still be prohibitive when this is not the case – e.g., when running on a UAV or a low-powered vehicle not able to support power-hungry hardware.

In a nutshell, marrying good practices to achieve generalization with online adaptation is essential for facing the real world, but still not sufficient when computational resources cannot support additional overhead at deployment time. In this context, the adaptation process has always been approached as a *single-instance* task, in which a single stereo network is deployed in an unseen environment and is gradually optimized over it. This setting ignores the existence of other instances of the model operating in different environments, potentially adapting independently to the specific domain they face. In a world where cameras and sensors are increasingly widespread, and fleets of autonomous vehicles are on the horizon, we argue adaptation itself can be formulated as a distributed task. In this scenario, an agent lacking sufficient computational capacity can demand the adaptation process to a network of peers equipped with more powerful hardware and thus capable of sustaining the adaptation process.

In this paper, we introduce a novel framework implementing *federated online adaptation* for deep stereo networks, by building on principles of federated learning [36]. Since communication between nodes is strictly necessary to carry out adaptation in a distributed manner, a connection overhead is introduced to transfer data, proportional to the quantity of data itself. To minimize this overhead, we design an algorithm specifically tailored to reduce the data quantity exchanged between agents at the most, while maintaining the effectiveness of the overall adaptation process nearly unaltered. This is done by revising the MAD algorithm [55] to the federated setting and thus developing **FedMAD**. Our federated framework is extensively evaluated on several stereo datasets, such as KITTI [16], DrivingStereo [67], and DSEC [14], proving that federated adaptation can provide an equivalent or, in the most challenging scenarios, even greater accuracy improvement compared to single-device adaptation, as spotlighted in Fig. 1. To the best of our knowledge, our work represents the first attempt to deal with real-time adaptation through a federated approach, in particular in the field of self-adapting stereo networks [41, 54, 55]. Our main claims can be resumed as:

- We revise real-time adaptation frameworks [41, 55] to introduce recent advances in deep stereo concerning generalization and architectural design, realizing a new baseline that largely improves over prior works.
- We introduce a novel framework casting online adaptation as a *federated* process, allowing to free the single device from the computational overhead that is instead distributed among a number of peer devices.
- Since distributed adaptation introduces data traffic between nodes over the web, we propose FedMAD, an algorithm built upon our new baseline to reduce the amount of data exchanged between nodes with negligible impact on adaptation effectiveness.
- We evaluate our framework paired with multiple real-time stereo networks on a variety of datasets, supporting that federated adaptation performs comparably to single-node adaptation, and even better in challenging domains.

2. Related Work

We briefly review the literature relevant to our work.

Deep Stereo Matching. The stereo matching literature counts several hand-crafted algorithms [43] through the years, usually divided into local and global methods according to their structure and their speed/accuracy trade-off. In the last decade, deep learning has brought a paradigm shift into stereo matching, achieving more and more accurate results on standard benchmarks [42]. While the first steps in this field aimed at replacing individual modules of the conventional pipeline [43] with compact networks [32, 46, 47, 59], with DispNet [35] the end-to-end architectures rapidly conquered the main stage [6, 9, 22, 29, 48, 55, 57, 66, 68]. Most of the models can be broadly classified into 2D [29, 35, 55, 64] and 3D [6, 9, 22, 48, 57, 66, 68] architectures, with some exploiting transformers [19, 28].

In the last years, several works focused on improving the capability of stereo networks to generalize across different domains, for instance by reducing the gap between training on synthetic and testing on real images. The main approaches involved the use of hand-crafted matching functions or algorithms [2, 5], techniques to learn for more robust features [11, 28, 31, 69], or the generation of photorealistic data for training [58, 62]. Eventually, the most recent stereo architectures [24, 30, 63, 65] proved to be capable of strong generalization from synthetic to real images even without making use of any of the aforementioned strategies.

Self-supervised Stereo. To overcome the need for annotated data, self-supervised techniques have been developed to directly train stereo networks on unlabelled image pairs. The minimization of the photometric error [18] between the left and right images, with the latter being warped according to estimated disparity, is at the core of most approaches trained on unconstrained stereo pairs [74, 75] or videos [10, 23, 61]. An alternative strategy consists of obtaining pseudo-labels from either hand-crafted algorithms [53, 56] or other depth estimation networks [1, 49].

Real-time Adaptation for Stereo. Although synthetic datasets provide countless annotated data, the poor generalization capabilities of the stereo models developed at first led to the development of adaptation techniques to overcome the synthetic to real domain shift directly during deployment. As this demands the model to adapt in the absence of ground truth, photometric losses [54, 55] and pseudo labels [41, 60] have been employed.

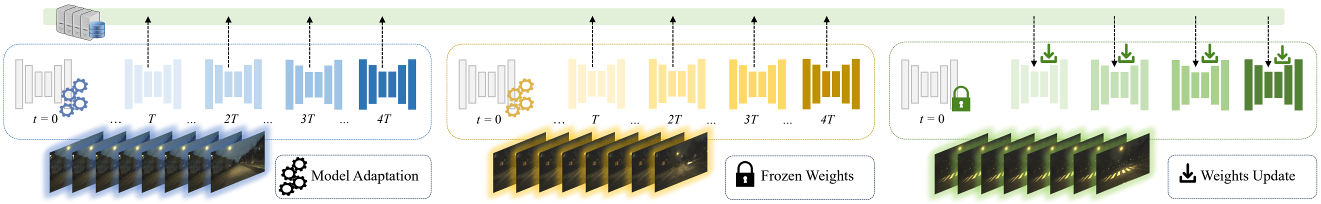

Figure 2. **Overview of our federated adaptation framework.** On the one hand, *active* nodes run online adaptation (blue and yellow) and periodically send their updated weights to a central server. On the other, a *listening* client (green) can benefit from the adaptation process carried out by the active nodes, by receiving aggregated weights updates from the server.

Federated Learning. This learning paradigm aims at training models from distributed data sources. A large body of literature has emerged in the last five years [26], mostly focusing on classification tasks. The pivotal federated learning algorithm is FedAvg [36]: a set of clients first train their local model using private data and then upload the weights to a server, where they are averaged to form a global model. Several methods [21, 25, 27, 37, 50, 51, 71] tried to regularize the local training phase in FedAvg [36], with FedProx [27] and SCAFFOLD [21] restricting the local update to be consistent globally, and MOON [25] applying a contrastive objective to regularize the optimization of local models to not deviate significantly from the global model. In contrast, personalized federated learning [7, 12, 13, 33, 50] aims at training custom models for each client to better fit local data. Finally, [8] shows the importance of exploiting pre-training when possible, as we do since we aim at deploying a distributed adaptation process.

3. Federated Adaptation for Deep Stereo

In this section, we introduce the basic principles over which our federated adaptation framework is developed.

3.1. Background: Online Adaptation for Stereo

Despite the recent advances in domain generalization [24, 30, 63, 65], a pre-trained stereo backbone might face drops in accuracy when deployed in challenging environments. As such, adapting the model online [41, 55] can be a solution for dealing with these occurrences. For any incoming stereo pair b_t, the network predicts a disparity map (or multiple, depending on the design) according to current weights w_t. Subsequently, it updates them by minimizing a loss function, typically the sum of multiple terms ℓ_i:

$$w_{t+1} \leftarrow w_t - \eta \nabla \sum_i \ell_i(w_t, b_t) \tag{1}$$

This step updates the whole set of parameters, thus carrying out *full adaptation* of the model (FULL) with non-negligible overhead – and consequent drop in framerate.

To mitigate this side effect, Tonioni *et al.* [55] introduced *Modular Adaptation* (MAD) along with a dedicated backbone (MADNet), made of 5 encoder-decoder blocks predicting disparity maps at different scales. For any adaptation step t, a block i is sampled according to a probability distribution, then only the corresponding output is used to compute the loss and optimize the subset of weights $w_t[i]$:

$$\begin{aligned} i &= \text{sample}(\text{softmax}(H)) \\ w_{t+1}[i] &\leftarrow w_t[i] - \eta \nabla \ell_i(w_t, b_t)[i] \end{aligned} \tag{2}$$

This significantly reduces the computational overhead required for adaptation. Consequently, MADNet coupled with MAD operates at double the speed compared to FULL, despite resulting in a moderate drop in accuracy.

Both strategies can be deployed using photometric losses [55] (FULL/MAD) or by leveraging proxy labels [41] when available (FULL++/MAD++).

3.2. Federated Adaptation Framework

The FULL and MAD algorithms are defined on a *single-instance* perspective – *i.e.*, a single stereo backbone is deployed and adapted during navigation. However, this paradigm alone might not be sufficient to overcome challenging domain changes or might be unusable if not supported by powerful enough hardware (*e.g.*, when the stereo models run on embedded devices, barely granting real-time processing even in the absence of any adaptation process).

Purposely, we design a federated framework in which we define a set of *active* nodes A, capable of adapting independently, and other *listening* clients C which demand the adaptation process to the former, as sketched in Figure 2. The two categories are managed by a central server, in charge of receiving updated weights and distributing them to the listening nodes. Algo. 1 defines the operations carried out by the server and the active clients. The server runs a loop (lines 4-14) during which it waits for updated weights transmitted by the active clients (lines 5-7). Once it has received the updates from each active client, the server aggregates such updates by computing the average of the weights as in FedAvg [36] and dispatches the updated model to clients C (lines 8-11). Clients A send their updates periodically after they perform T steps of adaptation (lines 15-19). We dub this framework *FedFULL*.

Algorithm 1 Federated Adaptation framework.

Server executes:

set $t = 0$, load pre-trained $w_t = w_0$
register adapting clients A, listening clients C
initialize buffers $W = [\,][\,]$, $H = [\,][\,]$
while True **do**
  for each client $k \in A$ **in parallel do**
    $W[k] \leftarrow \text{ClientUpdate}(k, w_t^k, T)$
  end for
  for each block i in w_t **do**
    $w_{t+1}[i] \leftarrow \frac{1}{||A||} \sum_{k \in A} W[k][i]$
    send w_{t+1} to C
  end for
  flush buffer $W = [\,]$
  $t \leftarrow t + 1$
end while

ClientUpdateFULL(k, w^k, T)**:** *// extends ClientUpdate*

for each step τ from 0 to T **do**
  sample batch b_τ
  update weights $w^k \leftarrow w - \eta \nabla \sum_i \ell_i(w^k, b_\tau)$
end for
return w^k to server

Algorithm 2 Modular Adaptation update.

ClientUpdateMAD(k, w^k, T, H)**:** *//extends ClientUpdate*

1: **for** each step τ from 0 to T **do**
2:   sample batch b_τ
3:   update weights $w^k \leftarrow w - \eta \nabla \sum_i \ell_i(w^k, b_\tau)$
4:   **for** each block i in H **do**
5:     $H[k][i]$ += 1 **if** i was updated
6:   **end for**
7: **end for**
8: $j \leftarrow \text{sample}(\text{softmax}(H[k]))$
9: $H[k][j] = 0.9 \cdot H[k][j]$
10: return $j, w^k[j]$ to server

This way, C receive updates to their weights and improve their accuracy, without actively running any GPU-intensive extra computation. However, significant data traffic between A, the server, and C is introduced, proportional to the number of parameters in the stereo network, the number of clients, and the updates interval T. Purposely, we propose a variant of the aforementioned federated framework inspired by MAD [55], by changing the updating procedure carried out by nodes A as outlined in Algo. 2. At each adaptation step, the client keeps track of the blocks it updates (lines 4-6) which could be some or all of them. Then, it samples a single block according to a probability distribution of the most updated blocks (line 8), sends it solely to the server, and decays its number of updates (line 9). On the server side, averaging is performed only for the subset of blocks received. We refer to this variant as *FedMAD*; we will show how it can reduce data traffic significantly, with a marginal drop in the accuracy of clients C.

3.3. Proposed Backbone: MADNet 2

With our federated framework being defined, we now select the stereo backbone to be coupled with it. MADNet [41, 55] would be a natural choice, since already designed to exploit modular adaptation and thus ready for both FedFULL and FedMAD variants. However, its accuracy, according to [41, 55], falls far behind the one achieved by modern state-of-the-art architectures [24, 30, 63, 65] and, despite the much higher efficiency, even while adapting, it cannot match their results.

Purposely, we revise the MADNet design and develop a new baseline for real-time self-adaptive deep stereo, which we dub MADNet 2. We argue that one of the weaknesses in its original architecture lies within the module responsible for building the cost volume at multiple scales. Specifically, it computes correlation scores between features along the epipolar line according to a radius r, defined as a hyper-parameter (the larger the radius, the higher the chance to hit the corresponding pixels) and collects them into coarse-to-fine volumes, processed by decoders to estimate disparity maps at different scales. For the sake of efficiency, small values of r are used – such as 2 as in the original MADNet – thus constraining the search range and, potentially, reducing accuracy for disparities falling out of it, despite the use of features warping at each scale.

We replace this module with the all-pairs correlation volume proposed by RAFT-Stereo [30], thus extending the search range to the entire epipolar line at any scale. Then, a pyramid of correlation scores is sampled and forwarded to the decoders: this ensures obtaining a fixed amount of channels as input to the decoder, independently of the image resolution. Differently from RAFT-Stereo, which builds a single volume at quarter resolution and iterates an arbitrary amount of times to estimate disparity, we build multiple volumes at lower scales (from $\frac{1}{64}$ up to $\frac{1}{4}$ as in the original MADNet) and estimate a fixed number of disparity maps. This, together with the very compact design of the entire architecture, trades the high accuracy achieved by RAFT-Stereo with a significantly lower running time (about $60\times$ lower). Finally, in our revised design we remove the context network [55] to further prioritize efficiency.

4. Experimental Results

In this section, we evaluate the impact of our framework.

4.1. Experimental Settings

Implementation Details. We implement our framework in PyTorch. We use models provided by the authors when available, or retrain them following the recommended settings – *e.g.*, we retrain those showing bad generalization

		City		Residential		Campus$^{(\times 2)}$		Road		All		Runtime	
Model	Adapt. mode	D1-all (%)	EPE (px)	D1-all (%)	EPE (px)	D1-all (%)	EPE (px)	D1-all (%)	EPE (px)	D1-all (%)	EPE (px)	3090 (ms)	AGX (ms)
RAFT-Stereo [30]	No adapt.	1.55	0.89	1.77	0.82	2.53	0.89	1.77	0.85	1.75	0.84	333	> 2000
CREStereo [24]		1.87	0.99	1.71	0.89	3.21	1.07	2.00	0.89	1.82	0.91	470	
IGEV-Stereo [63]		2.26	1.00	2.56	0.94	3.01	0.99	2.52	0.96	2.51	0.96	493	
UniMatch [65]		2.66	1.13	3.20	1.10	3.10	1.13	2.26	1.08	2.97	1.10	110	
MADNet [55]	No adapt.	37.42	9.96	37.04	11.34	51.98	11.94	47.45	15.71	38.84	11.68	7	64
MADNet 2 **(ours)**		4.04	1.10	4.05	1.03	6.07	1.29	4.01	1.08	4.21	1.09	5	47

(a) No adaptation – pre-trained on [35]

Model	Adapt. mode	City D1-all (%)	City EPE (px)	Residential D1-all (%)	Residential EPE (px)	Campus D1-all (%)	Campus EPE (px)	Road D1-all (%)	Road EPE (px)	All D1-all (%)	All EPE (px)	3090 (ms)	AGX (ms)
MADNet [55]	FULL	3.35	1.11	2.38	0.94	10.62	1.78	2.72	1.04	2.43	0.95	38	630
	MAD	7.51	1.63	4.37	1.32	22.27	3.66	9.38	2.04	4.09	1.19	15	121
MADNet 2	FULL	1.32	0.87	1.20	0.80	3.45	1.21	1.09	0.81	1.25	0.83	33	526
	MAD	1.40	0.88	1.20	0.81	3.84	1.15	1.11	0.80	1.26	0.84	11	80

(b) Adaptation – photometric loss [55]

Model	Adapt. mode	City D1-all (%)	City EPE (px)	Residential D1-all (%)	Residential EPE (px)	Campus D1-all (%)	Campus EPE (px)	Road D1-all (%)	Road EPE (px)	All D1-all (%)	All EPE (px)	3090 (ms)	AGX (ms)
MADNet [41]	FULL++	3.51	1.12	2.27	0.94	9.69	1.63	3.18	1.05	2.28	0.95	21	553
	MAD++	4.12	1.18	3.31	1.04	11.24	1.76	5.32	1.22	2.46	0.98	12	97
MADNet 2	FULL++	1.23	0.90	1.05	0.80	2.39	0.92	1.02	0.83	1.06	0.82	18	464
	MAD++	1.39	0.93	1.16	0.83	2.88	1.00	1.14	0.85	1.16	0.84	8	70

(c) Adaptation – proxy labels [41]

Table 1. **Online adaptation within a single domain.** Results on the *City*, *Residential*, *Campus*, and *Road* sequences from KITTI [17].

performance, using the augmentation strategy suggested in [58]. Federated runs are carried out on a server featuring 4× 3090 GPUs and AMD EPYC 7452 32-Core CPU. Each client runs independently on a single GPU, on a dedicated thread started through the Python threading module to enable concurrency. Unless otherwise specified, the listening client is supported by three clients running full adaptation, with update rate $T = 10$. To reduce the randomness due to allocation and run of any thread, the listening client starts only after other clients have started and transmitted their first update to the server. Then, they loop through their sequence until the listening client has fully processed its own. Regarding adaptation, we use FULL and MAD strategies from [55], whereas, for the former, we compute losses for any predicted disparity rather than for the latest only [55].

Evaluation Protocol. We follow [41, 55] to evaluate any model: we process the stereo pairs in a sequential order, mimicking an online acquisition scenario. We measure the D1-all error rate as the percentage of pixels having absolute disparity error larger than 3 and relative error larger than 5%, as well as the End-Point-Error (EPE). In the case of an adaptation, the error is computed before weights are updated. When performing federated adaptation, the active clients run on sequences from different domains to avoid any data leak and favor the passive client. We also report model speed by measuring the CUDA total execution time with PyTorch profiling tools – *i.e.*, not considering input/output overheads – both when running on nVidia RTX 3090 (350W consumption) or on a Jetson AGX Xavier embedded board (set in MAXN mode and consuming 30W), averaged over 100 runs after a bootstrap of 100 inferences. In most tables, we highlight the **best** and **second best** results among macro-categories.

4.2. Datasets

FlyingThings3D. A collection of synthetic images, comprising approximately 22k training stereo pairs with dense ground truth labels, part of the SceneFlow synthetic dataset [35]. Following [52], this dataset has been used to pre-train our model and other real-time networks.

KITTI [17]. A large dataset featuring 61 stereo sequences, for a total of about 43k pairs with 375×1242 average resolution. Following [41, 55], we test on *Road*, *Residential*, *Campus* and *City* domains obtained by concatenating all the sequences according to their classification on the official website, using filtered LiDAR measurements [15] converted to disparities as ground-truths.

DrivingStereo [67]. This dataset collects about 170k stereo images grouped in 38 sequences with an average resolution of 400×880 pixels. As defined in [41], we select the same *Rainy*, *Dusky*, and *Cloudy* sequences for evaluation. For federated experiments, we sample additional sequences according to their classification in [67], respectively tagged as *Foggy* (*2018-10-17-14-35*, *2018-10-22-10-44* and *2018-10-25-07-37*, since no other rain sequences are present on the dataset), *Dusky* (*2018-10-16-07-40*, *2018-10-16-11-13* and *2018-10-16-11-43*) and *Cloudy* (*2018-10-17-14-35*, *2018-10-17-15-38* and *2018-10-18-10-39*).

DSEC [14]. A dataset collected by means of stereo RGB and event cameras, providing 53 sequences for a total of about 50k stereo pairs at 1080×1440 resolution, for half of which ground-truth disparity is provided. From this dataset, we select four sequences to test online adaptation on nighttime images: *zurich_city_03_a*, *zurich_city_09_a*, *zurich_city_10_a* and *zurich_city_10_b*, respectively tagged as *Night#1*, *Night#2*, *Night#3* and *Night#4*. In federated experimets, we use sequences *zurich_city_09_b*, *zurich_city_09_c*, *zurich_city_09_d* and *zurich_city_09_e* for adapting active clients.

4.3. Evaluation on KITTI

Single-agent Adaptation. Tab. 1 collects the results achieved by several pre-trained stereo models on the single domains of KITTI. On top (a), we report state-of-the-art models [24, 30, 63, 65] characterized by outstanding generalization performance on this dataset, yet far from running in real-time – or even far from achieving 1 FPS on AGX – followed by MADNet and MADNet 2. The latter, although

Model	Fed. mode	City D1-all (%)	City EPE (px)	Residential D1-all (%)	Residential EPE (px)	Campus$^{(\times 2)}$ D1-all (%)	Campus$^{(\times 2)}$ EPE (px)	Road D1-all (%)	Road EPE (px)	Data Traffic To Server (MB/s)	Data Traffic To Client (MB/s)	Runtime 3090 (ms)	Runtime AGX (ms)
MADNet 2	FedFULL	1.42	0.89	1.22	0.80	3.93	1.14	1.12	0.80	20.2	6.6	5	47
	FedMAD	1.48	0.90	1.29	0.81	4.05	1.17	1.16	0.82	4.3	3.6		
	FedDEC	1.43	0.90	1.24	0.82	3.92	1.12	1.14	0.81	14.3	4.7		
	FedLAST	2.72	1.07	2.90	1.02	4.53	1.21	2.23	0.97	2.4	0.8		
	FedENC	3.44	1.05	3.40	0.98	5.56	1.23	3.43	1.03	6.8	2.3		
(a) Federated Adaptation – photometric loss [55]													
MADNet 2	FedFULL++	1.38	0.94	1.12	0.81	3.45	1.10	1.11	0.85	28.4	9.4	5	47
	FedMAD++	1.46	0.95	1.20	0.83	3.55	1.11	1.19	0.87	6.4	5.2		
	FedDEC++	1.54	0.98	1.35	0.86	3.74	1.11	1.22	0.90	20.1	6.7		
	FedLAST++	3.09	1.16	3.07	1.05	4.80	1.24	2.54	1.06	3.7	1.2		
	FedENC++	3.34	1.04	3.16	0.95	5.54	1.22	3.29	1.02	10.0	3.3		
(b) Federated Adaptation – proxy labels [41]													

Table 2. **Federated adaptation with MADNet 2.** Results on the *City*, *Residential*, *Campus*, and *Road* sequences from KITTI [17].

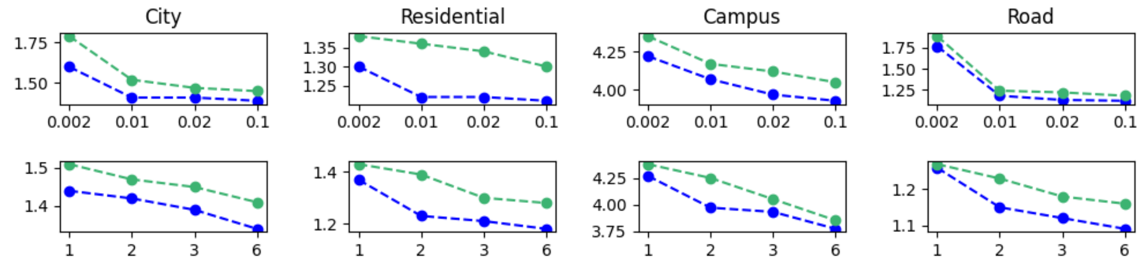

Figure 3. **Ablation study – impact of the update frequency (top) and number of agents (bottom) on accuracy.** We report D1-all (%) on the KITTI dataset for FedFULL (blue) and FedMAD (green).

generalizing largely better than the former, cannot reach the previous models yet, despite being unquestionably more efficient. Then, we report in (b) and (c) the results achieved by enabling adaptation using photometric loss [55] or proxy labels [41], either with FULL or MAD strategies [55]. In the latter case, we can observe slightly lower processing time, probably caused by the different effort required to compute the loss on sparse labels rather than reprojecting images and measuring photometric dissimilarity densely. Notably, MADNet falls short of achieving the accuracy of state-of-the-art models [24, 30, 63, 65] trained on synthetic data solely, even when adapting. Conversely, MADNet 2 largely benefits from its improved generalization. By enabling adaptation, it bridges the gap with state-of-the-art networks, even outperforming them when proxy labels are available [41], and still running in real-time on high-end hardware – while on lower-powered platforms it reaches nearly 15 FPS in its most efficient setup, *i.e.* MAD++, if dedicated hardware is available to get proxy labels [41].

Federated Adaptation. We now measure the boost in accuracy MADNet 2 gains when exploiting distributed adaptation. Tab. 2 reports the outcome of this experiment: for a client running on a domain, three remote clients adapt on 5 random sequences sampled from the other domains according to FULL (a) or FULL++ (b) algorithms. With reference to Tab. 1, we can notice how FedFULL/FedFULL++ consistently outperforms MAD/MAD++ (except on *Campus*), while not adding any computational overhead, thanks to the efforts by the distributed clients, yet at the cost of introducing some data traffic between nodes. This latter can be reduced by FedMAD – or some alternative strategies, consisting of averaging only the weights of the decoders

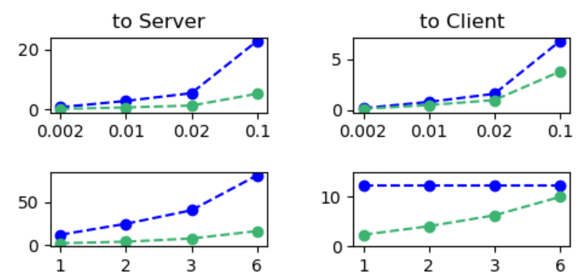

Figure 4. **Ablation study – impact of the update frequency (top) and number of clients (bottom) on traffic.** We report MB/s (top) and MB/updates (bottom) exchanged on the KITTI dataset for FedFULL (blue) and FedMAD (green).

(FedDEC), the last decoder (FedLAST), or the encoders [12] (FedENC) – while dampening the effect of adaptation. Only the former two nearly preserve the accuracy yielded by FedFULL, with FedMAD reducing the data traffic much more than FedDEC while also retaining the highest accuracy when the adapting clients mounting dedicated hardware to compute proxy labels. In this latter case, the listening client benefits from the boost given by labels yet without having any hardware dedicated to their computation.

Ablation Studies. The effectiveness of federated adaptation scales mainly with two hyper-parameters: i) the frequency at which each client pushes its updated model to the server, and ii) the number of remote clients actively contributing to adaptation. Both dictate the speed at which a passive agent will benefit from adaptation, as well as the volume of data being transferred to the cloud.

Fig. 3 examines the impact of both factors on accuracy with FedFULL and FedMAD. On top, we can observe how sending updates to the server once every 100 adaptation

Model	Adapt. mode	City D1-all (%)	City EPE (px)	Residential D1-all (%)	Residential EPE (px)	Campus$^{(\times 2)}$ D1-all (%)	Campus$^{(\times 2)}$ EPE (px)	Road D1-all (%)	Road EPE (px)	Data Traffic To Server (MB/s)	Data Traffic To Client (MB/s)	Runtime 3090 (ms)	Runtime AGX (ms)
CoEX [3]	No Adapt.	2.57	1.04	2.51	0.96	3.97	1.25	2.98	1.02	-	-	19	177
	FULL	0.93	0.81	0.79	0.72	2.11	0.90	0.85	0.77	-	-	80	1403
	FedFULL	1.13	0.84	0.90	0.74	2.55	0.99	1.12	0.80	8.2	2.2	19	177
HITNet [52]	No Adapt.	1.99	1.00	2.15	0.93	3.11	1.06	2.07	0.95	-	-	36	404
	FULL	0.92	0.81	0.93	0.74	2.15	0.88	0.83	0.76	-	-	110	1653
	FedFULL	0.94	0.82	0.94	0.74	2.03	0.82	0.90	0.79	2.2	0.6	36	404
TemporalStereo [72]	No Adapt.	4.33	1.26	3.47	1.10	3.80	1.19	4.67	1.21	-	-	42	✗
	FULL	1.06	0.82	0.99	0.76	2.90	1.03	0.87	0.75	-	-	162	✗
	FedFULL	1.25	0.86	1.04	0.78	2.24	0.91	1.15	0.82	31.2	8.9	42	✗

Table 3. **Online adaptation by fast networks (TemporalStereo [72], HITNet [52], CoEX [3]) within a single domain – single agent vs federated adaptation.** Results on the *City*, *Residential*, *Campus*, and *Road* sequences from KITTI [17].

Model	Adapt. mode	Rainy D1-all (%)	Rainy EPE (px)	Dusky D1-all (%)	Dusky EPE (px)	Cloudy D1-all (%)	Cloudy EPE (px)	Data Traffic To Server (MB/s)	Data Traffic To Client (MB/s)	Runtime 3090 (ms)	Runtime AGX (ms)
RAFT-Stereo [30]	No Adapt.	11.52	1.59	3.08	0.88	4.18	1.02	-	-	264	> 1000
CREStereo [24]		17.43	3.61	7.08	1.23	4.08	1.07	-	-	415	
IGEV-Stereo [63]		11.70	1.85	3.57	0.95	5.27	1.26	-	-	389	
UniMatch [65]		14.84	2.69	7.51	1.27	5.78	1.25	-	-	85	
CoEX [3]	No Adapt.	13.48	2.53	11.00	1.58	4.46	1.16	-	-	16	130
HITNet [52]		14.08	2.74	8.88	1.37	4.17	1.14	-	-	29	311
TemporalStereo [72]		18.53	3.94	13.61	1.80	6.02	1.31	-	-	33	✗
MADNet [55]	No Adapt.	27.14	3.90	24.73	2.45	11.00	1.77	-	-	6	64
MADNet 2 **(ours)**		16.47	3.03	13.16	1.66	6.72	1.35	-	-	4	43
(a) No Adaptation – pre-trained on [35]											
MADNet 2	FULL	10.19	1.70	11.36	1.54	5.76	1.27	-	-	30	492
	MAD	11.12	1.78	13.36	1.61	5.93	1.26	-	-	12	65
MADNet 2	FedFULL	11.57	2.00	10.65	1.44	5.45	1.20	20.6	6.8	4	43
	FedMAD	11.71	2.10	10.12	1.41	5.60	1.21	4.6	3.6	4	43
(b) Single-agent vs Federated Adaptation – photometric loss [55]											
MADNet 2	FULL++	10.34	2.27	4.41	1.04	5.20	1.63	-	-	20	470
	MAD++	10.06	2.01	5.25	1.09	4.34	1.09	-	-	8	48
MADNet 2	FedFULL++	8.33	1.73	4.13	1.00	4.55	1.13	28.8	9.6	4	43
	FedMAD++	8.58	1.74	4.40	1.01	4.65	1.16	6.5	4.5	4	43
(c) Single-agent vs Federated Adaptation – proxy labels [41]											

Table 4. **Online adaptation on DrivingStereo [67].** Results on the *Rainy*, *Dusky* and *Cloudy* sequences as selected in [41].

steps yields noticeable improvements in most cases already, saturating when increasing it to one every 10. At the bottom, we show how increasing the number of active clients consistently improves the results for the listening node.

Fig. 4 reports the amount of data transmitted from adapting clients to the server (left), as well as from the server to the listening client (right) as functions of the update frequency (top) and the number of clients (bottom). We highlight how FedMAD enables moderate growth in data traffic when the frequency is increased compared to FedFULL, with significant savings on the updates sent to the server. The gap with FedFULL becomes larger when more clients contribute to the process. In contrast, the data transferred to the listening client remains constant with FedFULL, and the saving by FedMAD nullifies beyond 6 clients.

Federated Adaptation – Other Networks. Online adaptation can be performed by any stereo network and, as such, federated adaptation can as well. Purposely, we implement FULL and FedFULL with other real-time stereo networks – CoEX [3], HITNet [52], and TemporalStereo [72] and evaluate their performance on KITTI. Tab. 3 collects the outcome of this experiment. We can notice how the three models can effectively adapt on the single domains, at the cost of dropping their efficiency. By demanding the adaptation process to distributed clients, all of them can benefit from an equivalent boost in performances while avoiding efficiency drops – with TemporalStereo and HITNet improving even more with FedFULL compared to FULL on *Campus*. Although CoEX and HITNet are slightly more accurate than MADNet 2 with reference to Tabs. 1 and 2, it is worth observing how both of these models require more than one second to generate a disparity map on AGX, and barely reach 5 FPS when adaptation is not enabled[1]. As such, we feel MADNet 2 is a more flexible solution for deploying real-time adaptive stereo systems, running at 20 FPS on AGX while federally adapting.

Additional Results. For the sake of space, we refer the reader to the **supplementary material** for more results.

4.4. Evaluation on DrivingStereo

Following [41], we evaluate our framework on the DrivingStereo dataset, characterized by more challenging environmental conditions harming the generalization capability of deep stereo models. Tab. 4 collects the results achieved by state-of-the-art models [24, 30, 63, 65], real-time networks [3, 52, 72], MADNet [55] and our MADNet 2 trained on synthetic data (a). We can notice how, in general, the error metrics are higher compared to those observed on KITTI, confirming the more challenging nature of this dataset. Again, MADNet 2 proves to generalize much better than MADNet, yet falls far behind the top-performing stereo networks, close to other fast models.

Adapting with photometric losses (b) within single domains only marginally improves the results on this benchmark – especially on *Dusky*: we ascribe this to the more challenging conditions depicted in these sequences, which

[1] we could not run [72] because of broken dependencies on AGX

Model	Adapt. mode	Night #1 D1-all (%)	Night #1 EPE (px)	Night #2 D1-all (%)	Night #2 EPE (px)	Night #3 D1-all (%)	Night #3 EPE (px)	Night #4 D1-all (%)	Night #4 EPE (px)	Data Traffic To Server (MB/s)	Data Traffic To Client (MB/s)	Runtime 3090 (ms)	Runtime AGX (ms)
RAFT-Stereo [30]	No adapt.	13.04	3.41	21.64	4.26	10.91	1.91	10.07	1.68	-	-	1030	> 8000
CREStereo [24]		11.34	2.38	23.48	3.19	15.37	2.39	12.42	1.75	-	-	1242	
IGEV-Stereo [63]		9.14	1.85	11.97	1.96	12.65	2.01	10.01	1.66	-	-	1250	
UniMatch [65]		34.29	5.43	39.80	5.32	26.75	3.29	26.29	3.28	-	-	480	
CoEX [3]	No adapt.	6.26	1.72	10.81	1.87	8.60	1.64	8.31	1.53	-	-	53	539
HITNet [52]		6.49	1.54	9.57	1.71	8.28	1.62	7.88	1.47	-	-	112	1400
TemporalStereo [72]		7.17	1.68	10.22	1.92	8.66	1.62	8.40	1.49	-	-	118	✗
MADNet 2 **(ours)**	No Adapt.	8.94	1.97	13.86	2.32	10.63	1.83	10.55	1.69	-	-	12	111
(a) No adaptation – pre-trained on [35]													
MADNet 2	FULL	5.65	1.41	9.16	1.60	8.12	1.50	8.97	1.46	-	-	102	1238
	MAD	5.79	1.52	8.87	1.60	7.89	1.49	8.50	1.46	-	-	30	253
MADNet 2	FedFULL	5.50	1.43	8.36	1.52	7.63	1.48	7.57	1.37	13.8	4.6	12	111
	FedMAD	5.52	1.43	8.39	1.53	7.91	1.50	7.79	1.39	2.9	2.0	12	111
(b) Single-agent vs Federated Adaptation – photometric loss [55]													
MADNet 2	FULL++	4.69	1.28	7.13	1.43	6.20	1.35	6.06	1.27	-	-	45	808
	MAD++	5.66	1.43	7.76	1.49	6.57	1.39	6.47	1.30	-	-	16	172
MADNet 2	FedFULL++	4.99	1.33	7.03	1.41	6.43	1.37	6.18	1.28	21.7	7.1	12	111
	FedMAD++	4.99	1.34	7.13	1.42	6.48	1.38	6.23	1.28	7.3	5.8	12	111
(c) Single-agent vs Federated Adaptation – proxy labels [41]													

Table 5. **Online adaptation on DSEC [14].** Results on the *Night#1*, *Night#2*, *Night#3* and *Night#4* sequences.

potentially compromise the effectiveness of the photometric loss. In these conditions, the possibility of relying on the adaptation carried out by other clients results crucial also in terms of accuracy, allowing both FedFULL and FedMAD to achieve better results on *Dusky* and *Cloudy* compared to standard FULL/MAD executed over the two domains.

When proxy labels are available (c), both FULL++ and MAD++ produce notably better results, yet leveraging the adaptation carried out remotely with FedFULL++ and FedMAD++ allows to improve the results even further on *Rainy* and *Dusky* – on the former in particular, it gains about 1.5% in D1-all – while resulting comparable on *Cloudy*.

In summary, a client demanding adaptation to the cloud can benefit even more than carrying it out independently in challenging environments, while avoiding runtime overheads. Accordingly, MADNet 2 can still run in real-time on AGX and surpass other fast models [3, 52, 72], running not even at 10 FPS there. The **supplementary material** reports federated experiments with other real-time models.

4.5. Evaluation on DSEC

We conclude by running further experiments on nighttime stereo sequences taken from the DSEC dataset [14]. Tab. 5 collects the results yielded by any stereo model considered so far on four selected night sequences. In contrast to KITTI and DrivingStereo, in (a) we can notice how the state-of-the-art models achieve a much higher error rate, with real-time architectures proving to be more robust in this context. Moreover, the higher resolution of this dataset makes the runtime of each method increase notably, with MADNet 2 being the only model still capable of retaining almost 10 FPS on AGX when not adapting. By actively adapting with FULL or MAD (b,c), MADNet 2 can further improve its accuracy and outperform the other stereo models, while dropping below 5 FPS. In such a setting, we can further appreciate how FedFULL becomes crucial for maintaining reasonable runtime while enjoying the benefits of adaptation, outperforming MAD either when using photometric loss (b) or proxy labels (c), and even being more effective than FULL in the former case. Given the lower inference speed caused by the dataset resolution, we can notice lower data traffic. This occurs as the adapting models require more time to perform the T steps set to update the server. Yet, FedMAD still allows for further reducing the communication overhead with little drops in accuracy.

For the sake of completeness, in the **supplementary material** we report the results by other real-time models.

5. Conclusion

In this paper, we presented for the first time a framework that implements federated online adaptation for deep stereo models. By demanding the optimization process to distributed nodes, a single model can benefit from adaptation even when deployed on low-powered hardware, thus improving its accuracy while maintaining its original processing speed. This achievement comes at the cost of introducing data traffic between nodes; however, this traffic can be reduced by means of an appropriate strategy that updates only some portions of the entire model at each communication round, specifically tailored for our MADNet 2. Exhaustive experiments showcase the effectiveness of our framework and its ability to be combined with different models.

Limitations. At now, passive clients benefit from the adaptation carried out by some active clients, without the latter receiving reciprocal benefits in return, and the adapting nodes process images with similar properties (resolution, depth range, application context) to those observed by the listening client. Finally, as clients run on the same server, connection delays are ignored in our experiments.

Future Work. We foresee federated adaptation will be applied to other visual tasks for which online adaptation is a reality, such as single image depth estimation [73], optical flow [39] or semantic segmentation [4, 40].

Acknowledgment. We thank LAR Laboratory (University of Bologna) for providing the Jetson AGX Xavier.

References

[1] Filippo Aleotti, Fabio Tosi, Li Zhang, Matteo Poggi, and Stefano Mattoccia. Reversing the cycle: self-supervised deep stereo through enhanced monocular distillation. In *16th European Conference on Computer Vision (ECCV)*. Springer, 2020.

[2] Filippo Aleotti, Fabio Tosi, Pierluigi Zama Ramirez, Matteo Poggi, Samuele Salti, Stefano Mattoccia, and Luigi Di Stefano. Neural disparity refinement for arbitrary resolution stereo. In *International Conference on 3D Vision*, 2021. 3DV.

[3] Antyanta Bangunharcana, Jae Won Cho, Seokju Lee, In So Kweon, Kyung-Soo Kim, and Soohyun Kim. Correlate-and-excite: Real-time stereo matching via guided cost volume excitation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021.

[4] Marc Botet Colomer, Pier Luigi Dovesi, Theodoros Panagiotakopoulos, Joao Frederico Carvalho, Linus Härenstam-Nielsen, Hossein Azizpour, Hedvig Kjellström, Daniel Cremers, and Matteo Poggi. To adapt or not to adapt? real-time adaptation for semantic segmentation. In *IEEE International Conference on Computer Vision*, 2023. ICCV.

[5] Changjiang Cai, Matteo Poggi, Stefano Mattoccia, and Philippos Mordohai. Matching-space stereo networks for cross-domain generalization. In *2020 International Conference on 3D Vision (3DV)*, pages 364–373, 2020.

[6] Jia-Ren Chang and Yong-Sheng Chen. Pyramid stereo matching network. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5410–5418, 2018.

[7] Hong-You Chen and Wei-Lun Chao. On bridging generic and personalized federated learning for image classification. In *ICLR*, 2022.

[8] Hong-You Chen, Cheng-Hao Tu, Ziwei Li, Han Wei Shen, and Wei-Lun Chao. On the importance and applicability of pre-training for federated learning. In *The Eleventh International Conference on Learning Representations*, 2022.

[9] 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. *Advances in Neural Information Processing Systems*, 33, 2020.

[10] Cheng Chi, Qingjie Wang, Tianyu Hao, Peng Guo, and Xin Yang. Feature-level collaboration: Joint unsupervised learning of optical flow, stereo depth and camera motion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2463–2473, 2021.

[11] WeiQin Chuah, Ruwan Tennakoon, Reza Hoseinnezhad, Alireza Bab-Hadiashar, and David Suter. ITSA: An information-theoretic approach to automatic shortcut avoidance and domain generalization in stereo matching networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13022–13032, 2022.

[12] Liam Collins, Hamed Hassani, Aryan Mokhtari, and Sanjay Shakkottai. Exploiting shared representations for personalized federated learning. In *ICML*, 2021.

[13] Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach. In *NeurIPS*, 2020.

[14] Mathias Gehrig, Willem Aarents, Daniel Gehrig, and Davide Scaramuzza. DSEC: A stereo event camera dataset for driving scenarios. *IEEE Robotics and Automation Letters*, 2021.

[15] Andreas Geiger, Martin Roser, and Raquel Urtasun. Efficient large-scale stereo matching. In *Asian conference on computer vision*, pages 25–38. Springer, 2010.

[16] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? The KITTI vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[17] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The KITTI dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.

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

[19] Weiyu Guo, Zhaoshuo Li, Yongkui Yang, Zheng Wang, Russell H Taylor, Mathias Unberath, Alan Yuille, and Yingwei Li. Context-enhanced stereo transformer. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022.

[20] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In *International conference on machine learning*, pages 1989–1998. Pmlr, 2018.

[21] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *ICML*, 2020.

[22] Alex Kendall, Hayk Martirosyan, Saumitro Dasgupta, Peter Henry, Ryan Kennedy, Abraham Bachrach, and Adam Bry. End-to-end learning of geometry and context for deep stereo regression. In *The IEEE International Conference on Computer Vision (ICCV)*, 2017.

[23] Hsueh-Ying Lai, Yi-Hsuan Tsai, and Wei-Chen Chiu. Bridging stereo matching and optical flow via spatiotemporal correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1890–1899, 2019.

[24] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16263–16272, 2022.

[25] Qinbin Li, Bingsheng He, and Dawn Song. Model-contrastive federated learning. In *CVPR*, 2021.

[26] Qinbin Li, Zeyi Wen, Zhaomin Wu, Sixu Hu, Naibo Wang, Yuan Li, Xu Liu, and Bingsheng He. A survey on federated learning systems: vision, hype and reality for data privacy and protection. *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[27] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. In *MLSys*, 2020.

[28] Zhaoshuo Li, Xingtong Liu, Nathan Drenkow, Andy Ding, Francis X Creighton, Russell H Taylor, and Mathias Unberath. Revisiting stereo depth estimation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6197–6206, 2021.

[29] Zhengfa Liang, Yiliu Feng, Yulan Guo, Hengzhu Liu, Wei Chen, Linbo Qiao, Li Zhou, and Jianfeng Zhang. Learning for disparity estimation through feature constancy. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[30] Lahav Lipson, Zachary Teed, and Jia Deng. RAFT-Stereo: Multilevel recurrent field transforms for stereo matching. In *International Conference on 3D Vision (3DV)*, 2021.

[31] Biyang Liu, Huimin Yu, and Guodong Qi. Graftnet: Towards domain generalized stereo matching with a broad-spectrum and task-oriented feature. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13012–13021, 2022.

[32] Wenjie Luo, Alexander G Schwing, and Raquel Urtasun. Efficient deep learning for stereo matching. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5695–5703, 2016.

[33] Xiaosong Ma, Jie Zhang, Song Guo, and Wenchao Xu. Layer-wised model aggregation for personalized federated learning. In *CVPR*, 2022.

[34] David Marr and Tomaso Poggio. Cooperative computation of stereo disparity: A cooperative algorithm is derived for extracting disparity information from stereo image pairs. *Science*, 194(4262):283–287, 1976.

[35] 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 *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[36] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *AISTATS*, 2017.

[37] Matias Mendieta, Taojiannan Yang, Pu Wang, Minwoo Lee, Zhengming Ding, and Chen Chen. Local learning matters: Rethinking data heterogeneity in federated learning. In *CVPR*, 2022.

[38] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[39] Chaerin Min, Taehyun Kim, and Jongwoo Lim. Meta-learning for adaptation of deep optical flow networks. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2145–2154, 2023.

[40] Theodoros Panagiotakopoulos, Pier Luigi Dovesi, Linus Härenstam-Nielsen, and Matteo Poggi. Online domain adaptation for semantic segmentation in ever-changing conditions. In *European Conference on Computer Vision (ECCV)*, 2022.

[41] Matteo Poggi, Alessio Tonioni, Fabio Tosi, Stefano Mattoccia, and Luigi Di Stefano. Continual adaptation for deep stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2021.

[42] Matteo Poggi, Fabio Tosi, Konstantinos Batsos, Philippos Mordohai, and Stefano Mattoccia. On the synergies between machine learning and binocular stereo for depth estimation from images: a survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[43] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *IJCV*, 47(1-3):7–42, 2002.

[44] 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 *German conference on pattern recognition*, pages 31–42. Springer, 2014.

[45] 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 *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3260–3269. IEEE, 2017.

[46] Akihito Seki and Marc Pollefeys. Patch based confidence prediction for dense disparity map. In *Proceedings of the British Machine Vision Conference 2016*, page 23. BMVC, 2016.

[47] Akihito Seki and Marc Pollefeys. SGM-Nets: Semi-global matching with neural networks. In *CVPR*, pages 231–240, 2017.

[48] Zhelun Shen, Yuchao Dai, and Zhibo Rao. Cfnet: Cascade and fused cost volume for robust stereo matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13906–13915, 2021.

[49] Zhelun Shen, Xibin Song, Yuchao Dai, Dingfu Zhou, Zhibo Rao, and Liangjun Zhang. Digging into uncertainty-based pseudo-label for robust stereo matching. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (01):1–18, 2023.

[50] Benyuan Sun, Hongxing Huo, Yi Yang, and Bo Bai. Partialfed: Cross-domain personalized federated learning via partial initialization. *NeurIPS*, 2021.

[51] Yue Tan, Guodong Long, Lu Liu, Tianyi Zhou, Qinghua Lu, Jing Jiang, and Chengqi Zhang. Fedproto: Federated prototype learning across heterogeneous clients. In *AAAI*, 2022.

[52] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14362–14372, 2021.

[53] Alessio Tonioni, Matteo Poggi, Stefano Mattoccia, and Luigi Di Stefano. Unsupervised adaptation for deep stereo. In *The IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2017.

[54] Alessio Tonioni, Oscar Rahnama, Tom Joy, Luigi Di Stefano, Ajanthan Thalaiyasingam, and Philip Torr. Learning to adapt for stereo. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2019.

[55] Alessio Tonioni, Fabio Tosi, Matteo Poggi, Stefano Mattoccia, and Luigi Di Stefano. Real-time self-adaptive deep stereo. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2019.

[56] Alessio Tonioni, Matteo Poggi, Stefano Mattoccia, and Luigi Di Stefano. Unsupervised domain adaptation for depth prediction from images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[57] Fabio Tosi, Yiyi Liao, Carolin Schmitt, and Andreas Geiger. Smd-nets: Stereo mixture density networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[58] Fabio Tosi, Alessio Tonioni, Daniele De Gregorio, and Matteo Poggi. Nerf-supervised deep stereo. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 855–866, 2023.

[59] Jure Žbontar, Yann LeCun, et al. Stereo matching by training a convolutional neural network to compare image patches. *J. Mach. Learn. Res.*, 17(1):2287–2318, 2016.

[60] Haiyang Wang, Xinchao Wang, Jie Song, Jie Lei, and Mingli Song. Faster self-adaptive deep stereo. In *Proceedings of the Asian Conference on Computer Vision*, 2020.

[61] Yang Wang, Peng Wang, Zhenheng Yang, Chenxu Luo, Yi Yang, and Wei Xu. Unos: Unified unsupervised optical-flow and stereo-depth estimation by watching videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8071–8081, 2019.

[62] Jamie Watson, Oisin Mac Aodha, Daniyar Turmukhambetov, Gabriel J. Brostow, and Michael Firman. Learning stereo from single images. In *European Conference on Computer Vision (ECCV)*, 2020.

[63] Gangwei Xu, Xianqi Wang, Xiaohuan Ding, and Xin Yang. Iterative geometry encoding volume for stereo matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21919–21928, 2023.

[64] Haofei Xu and Juyong Zhang. Aanet: Adaptive aggregation network for efficient stereo matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1959–1968, 2020.

[65] Haofei Xu, Jing Zhang, Jianfei Cai, Hamid Rezatofighi, Fisher Yu, Dacheng Tao, and Andreas Geiger. Unifying flow, stereo and depth estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[66] Gengshan Yang, Joshua Manela, Michael Happold, and Deva Ramanan. Hierarchical deep stereo matching on high-resolution images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5515–5524, 2019.

[67] Guorun Yang, Xiao Song, Chaoqin Huang, Zhidong Deng, Jianping Shi, and Bolei Zhou. Drivingstereo: A large-scale dataset for stereo matching in autonomous driving scenarios. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[68] Feihu Zhang, Victor Prisacariu, Ruigang Yang, and Philip HS Torr. GA-Net: Guided aggregation net for end-to-end stereo matching. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[69] Feihu Zhang, Xiaojuan Qi, Ruigang Yang, Victor Prisacariu, Benjamin Wah, and Philip Torr. Domain-invariant stereo matching networks. In *Europe Conference on Computer Vision (ECCV)*, 2020.

[70] Jiawei Zhang, Xiang Wang, Xiao Bai, Chen Wang, Lei Huang, Yimin Chen, Lin Gu, Jun Zhou, Tatsuya Harada, and Edwin R Hancock. Revisiting domain generalized stereo matching networks from a feature consistency perspective. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13001–13011, 2022.

[71] Lin Zhang, Li Shen, Liang Ding, Dacheng Tao, and Ling-Yu Duan. Fine-tuning global model via data-free knowledge distillation for non-iid federated learning. In *CVPR*, 2022.

[72] Youmin Zhang, Matteo Poggi, and Stefano Mattoccia. Temporalstereo: Efficient spatial-temporal stereo matching network. In *IROS*, 2023.

[73] Zhenyu Zhang, Stephane Lathuiliere, Elisa Ricci, Nicu Sebe, Yan Yan, and Jian Yang. Online depth learning against forgetting in monocular videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[74] Yiran Zhong, Yuchao Dai, and Hongdong Li. Self-supervised learning for stereo matching with self-improving ability. *arXiv:1709.00930*, 2017.

[75] Chao Zhou, Hong Zhang, Xiaoyong Shen, and Jiaya Jia. Unsupervised learning of stereo matching. In *The IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2017.