Supplementary Material: Camera Pose Estimation and Localization with Active Audio Sensing

Karren Yang[2⋆] Michael Firman[1] Eric Brachmann[1] Clément Godard[3⋆]

[1]Niantic [2]MIT [3]Google

A Implementation Details

In this section, we describe the implementation details for our pretraining task and our device localization tasks. All models are implemented and trained in PyTorch [14].

A.1 Pretraining Task

Input Representation. The network inputs are derived from 60ms of audio echo recordings (2 channels). For the Replica dataset [18], we sample the recordings at 44.1kHz and obtain magnitude spectrograms via STFT in Librosa [12] with 512 frequency bins and Hanning window of 64, following [6, 13]. For the Matterport3D dataset [3], we sample the recordings at 16kHz and obtain magnitude spectrograms with 512 frequency bins and Hanning window of 32, following [13]. In addition to magnitude spectrograms, we also compute the angle between the complex spectrograms of the two channels to take phase information into account. For the final input, we stack the spectrograms of the two audio channels with the angle information.

Audio Feature Extractor Network (f). The backbone of our audio feature extractor network is a ResNet18 model [8]. We input the audio as described above and extract features prior to the global pooling layer. The features are passed through a 1×1 convolution block with 32 output channels, ReLU activation and batch normalization. The resulting output is flattened to remove the time axis and processed through two more 1×1 convolution blocks to obtain 1D feature embeddings of sizes 512 and 512×6 respectively. Each length-512 subset of the latter embedding represents scene geometry from a different view along an egocentric cube map and is input to the depth decoder network separately.

Depth Decoder Network (g). The depth decoder takes length-512 audio feature embeddings as input and processes them through 7 up-convolution blocks to produce a 128×128 depth map with values between 0 and 1 (normalized by maximum depth of dataset), following [6, 13]. Each up-convolution block except the final one consists of nearest-neighbor up-sampling by a factor of 2, followed

⋆ Work done while at Niantic, during Karren's internship.

by a 3×3 convolution, ReLU activation and batch normalization. The final up-convolution block replaces the non-linear activation and batch normalization with a sigmoid activation. The number of output channels is halved with each of the first 6 blocks, starting at 512 and ending at 16. The final block outputs to one channel only for the depth map.

Training. Models are optimized for 300 epochs using Adam [21] with a learning rate of 1e-4, weight decay 5e-4, and batch size 64.

A.2 Relative Pose Estimation

Audio only model. The audio input representation and feature extractor network are the same as described in the pretraining task. The extracted features from f are put through a regression head, which is a shallow neural network with two hidden layers (2048 units) and ReLU activation. The final layer of the multi-layer consists of two linear heads, one for producing the 6-D representation of the rotation, and the other for producing the 3D representation of the translation. The audio extractor network is fixed with the weights from pretraining, and the rest of the model is trained using the loss described in the main paper with the hyperparameter β set to 50. We train the model for 300 epochs, using the Adam optimizer [21] with learning rate 1e-4, beta 0.9, weight decay 5e-4, and batch size 75.

Gating network. The gating network takes as input the 12-D pose outputs from the audio only model and the pretrained indoor SuperGlue model from [17] and processes them with a shallow neural network with two hidden layers (512 units) and ReLU activation. The final layer maps the output to a 4-D vector. Concretely, the output is a vector $z \in [0,1]^4, \sum_i z_i = 1$ indicating the composition of the final prediction,

$$\hat{R}, \hat{t} = \begin{cases} \hat{R}_a, \hat{t}_a & \text{if } \arg\max_i z_i = 0 \\ \hat{R}_v, \hat{t}_a & \text{if } \arg\max_i z_i = 1 \\ \hat{R}_a, \hat{t}_v & \text{if } \arg\max_i z_i = 2 \\ \hat{R}_v, \hat{t}_v & \text{if } \arg\max_i z_i = 3 \end{cases} \tag{1}$$

where the notation follows from the main paper. We train the gating network to minimize the cross-entropy loss between z and z^*, a one-hot vector indicating the optimal combination of expert outputs. We train for 100 epochs using the Adam optimizer [21] with learning rate 1e-4, beta 0.9, weight decay 5e-4, and batch size 75.

A.3 Place Recognition

Audio descriptor. The audio input representation and feature extractor network are the same as described in the pretraining task. The extracted features from f are put through a 1×1 convolution block with output size 128, the output of which is normalized to length 1. The audio feature extractor network is fixed

	Network	Weight sharing	Phase	RMS↓	REL↓	Log10↓	A1↑	A2↑	A3↑
	EchoNet [13]			0.561	0.449	0.142	0.596	0.766	0.855
	EchoNet [13]	✓		0.544	0.434	0.135	0.626	0.779	0.863
	ResNet18	✓		0.508	**0.397**	0.127	0.648	0.796	0.872
Ours ⇒	ResNet18	✓	✓	**0.502**	0.406	**0.126**	**0.653**	**0.798**	**0.873**

Table 1. Ablation Results for Surround Depth Estimation. Our weight sharing architecture and use of phase information improves over the baseline models. See text for details.

with the weights from pretraining, and the 1×1 block is trained on the triplet loss as described in the main paper with margin 0.5 for 100 epochs, using the Adam optimizer [21] with learning rate 1e-4, beta 0.9, weight decay 5e-4, and batch size 64. Subsequently, the entire model is finetuned for several epochs (1 for Replica, 10 for Matterport3D) with learning rate 1e-5.

Gating network. The gating network takes as input the 6-D pose outputs from SuperGlue matching [17] between from query image and retrieved image with NetVLAD [1] and processes them with a shallow neural network with two hidden layers (128 units) and ReLU activation. The final layer maps the output to a scalar value $z \in [0, 1]$ indicating whether to use the position retrieved by vision or audio:

$$\hat{c} = \begin{cases} c_{\mathrm{NN}(\mathbf{v}_q)} & \text{if } z > 0.5 \\ c_{\mathrm{NN}(\mathbf{y}_q)} & \text{otherwise} \end{cases} \tag{2}$$

where the notation follows that from the main paper. The gating function is trained to minimize the binary cross-entropy loss between z and z^*, which indicates whether the retrieved result from vision is better than audio, i.e., $z^* := \mathbb{1}_{||t_{\mathrm{NN}(v_q)} - t|| < ||t_{\mathrm{NN}(y_q)} - t||}$. We train for several hundred epoches (100 epochs for Replica scenes, 300 epochs for Matterport3D) using the Adam optimizer [21] with learning rate 1e-4, beta 0.9, weight decay 5e-4, and batch size 75.

A.4 Absolute Pose Regression

The audio input representation and feature extractor network are the same as described in the pretraining task. We fuse the outputs of the audio stream and the visual stream (256×256 RGB images processed using pretrained ResNet18 [8]) together using a transformer encoder module (3 layers, 8 heads). The fused features are put through a regression head, which is a shallow neural network with two hidden layers (2048 units) and ReLU activation. The final layer of the network consists of two linear heads, one for producing the 6-D representation of the rotation, and the other for producing the 3D representation of the translation. The audio extractor network is fixed with the weights from pretraining, and the rest of the model is trained using the loss described in the main paper with the hyperparameter β set to 50. We train the model for 300 epochs, using the Adam optimizer [21] with learning rate 2e-5, beta 0.9, weight decay 5e-4, and batch size 75.

	Overall		Correct Retrieval	
	Position Med. Error (m)	Rotation Med. Error (deg)	Position Med. Error (m)	Rotation Med. Error (deg)
Replica dataset				
Audio only				
– Regression	2.50	**15.7**	-	-
– Place recognition + relative pose	**0.61**	16.3	0.27	11.1
Matterport3D dataset				
Audio only				
– Regression	8.33	62.7	-	-
– Place recognition + relative pose	**2.21**	**48.0**	0.80	27.4

Table 2. Audio-only absolute pose estimation. We can perform absolute pose estimation by combining our audio-only place recognition and relative pose estimation models. Results are averaged over two Replica scenes and three Matterport3D scenes. See text for details.

B Additional Results

B.1 Pretraining Task: Ablation Results

Table 1 shows ablation results of our model on the Replica dataset. The first row refers to the model of [13] trained to predict each of the six faces of the cube, with the same size representation as our model; EchoNet refers their original feature extraction network. We observe improvements when we use our framework (with shared weights) to estimate depth from audio in different directions (compare rows 1 and 2, w/ and w/o weight sharing). Our architecture for the audio extraction network based on ResNet18 architecture [8] outperforms the original EchoNet (compare rows 2 and 3). Adding phase information as input further improves the performance of the model (compare rows 3 and 4).

B.2 Absolute Pose Estimation: Combining Retrieval with Relative Pose Estimation

Although we focus on absolute pose regression using an end-to-end network in the main paper, many recent vision pipelines for absolute pose estimation combine retrieval with relative pose estimation [11, 2, 22, 19, 20, 16, 9]. In Table 2, we show that an audio-only pipeline is also possible for this task by combining our approaches for place recognition and relative pose estimation. Correct retrieval refers to test examples where the outcome of place recognition falls within an accuracy threshold of 1m for Replica and 3m for Matterport3D. As expected, performance is better on the Replica scenes, which are smaller than the Matterport3D scenes, and depends heavily on whether the retrieval step is successful. This approach outperforms the audio-only regression baseline in the main paper, and could be integrated with two-step vision pipelines to make them robust to low overlap, poor illumination, etc. Note this audio-only method still does not outperform the proposed audio-visual method in the main paper.

Input	Position Med. Error (m)	Rotation Med. Error (deg)
Audio only		
– 10 SNR	1.08	8.2
– 30 SNR	0.80	6.4
– 50 SNR	0.72	5.8
– No Noise	0.74	6.4
Audio-Visual		
– 10 SNR	0.96	6.9
– 30 SNR	0.79	6.3
– 50 SNR	0.74	6.0
– No Noise	0.74	6.2

Table 3. Effect of Noise (Absolute Pose Regression).

	Standard Baseline (Relative Rot. <90°)			**Wide Baseline** (Relative Rot. >90°)			**Low Lighting** (Dark Image)		
	t↓	R↓	Acc.↑	t↓	R↓	Acc.↑	t↓	R↓	Acc.↑
Replica dataset									
Visual regression	35.0	15.0	21.0	37.4	20.0	17.3	57.3	99.1	0.7
AV regression	33.0	14.3	24.1	35.6	18.4	19.5	58.5	91.6	0.9
AV regression w/ attn fusion	17.8	8.7	46.2	**20.0**	10.7	**40.0**	32.9	26.9	15.4
Ours	**13.8**	**6.5**	**55.4**	22.6	**10.6**	37.6	**21.5**	**10.3**	**38.4**

Table 4. Additional Baselines: Relative Pose Estimation

B.3 Performance Under Noise

As real-world audio scenarios contain environmental sources of noise, we assessed the ability of audio echoes to capture geometric information in the presence of noise. Specifically, we performed absolute pose regression on one Matterport3D scene where the data is augmented with noise audio samples from the ACE Challenge 2015 [5]. In this setup, the train and test samples are captured from the same grid locations, but from different camera rotations. While performance degrades as the level of noise increases, as shown in Table 3, we note that the performance degradation is not substantial at 10 SNR.

B.4 Additional Baselines

In addition to the baselines in the main paper, we also experimented with the baselines shown in Supplemental Tables 4, 5, 6. For relative pose estimation, we considered a regression model that uses a transformer encoder [4] to fuse audio and visual signals. For place recognition, we considered AP-GeM [7, 15] as an additional visual baseline. For absolute pose regression, we follow the original implementation of PoseNet [10] by training a visual baseline using quaternion outputs. None of these approaches outperform the proposed audio-visual methods in the main paper.

	Overall All Queries		**High Overlap** Subset of Queries		**Low Overlap** Subset of Queries		**Low Lighting** All Queries	
	R@1	R@5	R@1	R@5	R@1	R@5	R@1	R@5
Replica dataset								
NetVLAD [1]	0.59	0.76	0.91	**0.98**	0.38	0.61	0.18	0.34
AP-GeM [7, 15]	0.54	0.71	0.86	0.96	0.32	0.55	0.19	0.40
Ours	**0.71**	**0.83**	**0.92**	**0.98**	0.58	**0.73**	0.64	**0.74**

Table 5. Additional Baselines: Place Recognition

	Overall All Queries		**Low Ambiguity** Normal Light		**High Ambiguity** Low Light	
	Position Error	Rotation Error	Position Error	Rotation Error	Position Error	Rotation Error
Replica dataset						
PoseNet w/ 6D output [10]	0.53	9.4	**0.43**	7.3	0.74	13.7
PoseNet w/ quat output [10]	0.52	12.8	0.44	9.7	0.69	19.2
Ours	**0.52**	**6.9**	0.46	**5.4**	**0.64**	**9.9**

Table 6. Additional Baselines: Absolute Pose Regression.

B.5 Relative Pose Estimation: Additional Qualitative Results

Figure 1 shows additional qualitative results of visual matching + audio + gating. In Fig. 1(a), the two devices have large relative rotation and there is low overlap between their images. As a result, visual matching performs poorly, as shown in Fig. 1(b). Our gating function chooses the audio expert to produce robust results, as shown in Fig. 1(c).

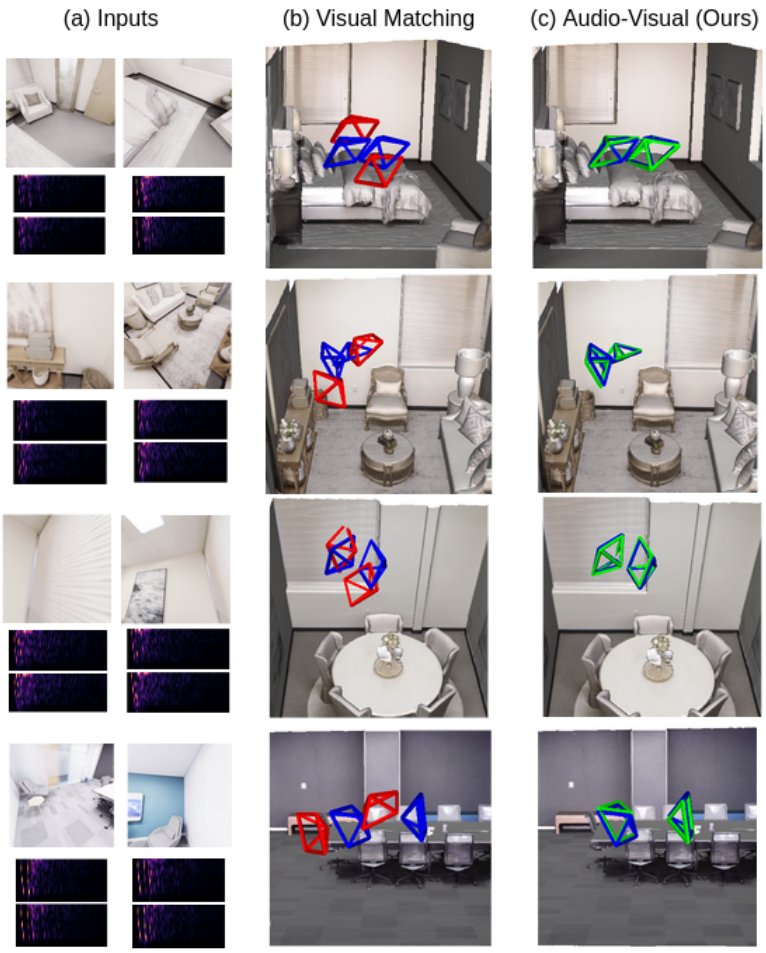

Fig. 1. Combining audio and vision for relative pose estimation. Blue – ground truth. Red – visual prediction. Green – our prediction. See text for details.

B.6 Place Recognition: Qualitative Results

Figure 2 shows additional qualitative results of our visual descriptor + audio + gating model. Since there is no visual overlap between the query sample (blue) and the images in the reference database, NetVLAD retrieval returns an incorrect result (red). The audio-visual model chooses the audio expert to make a correct retrieval (green).

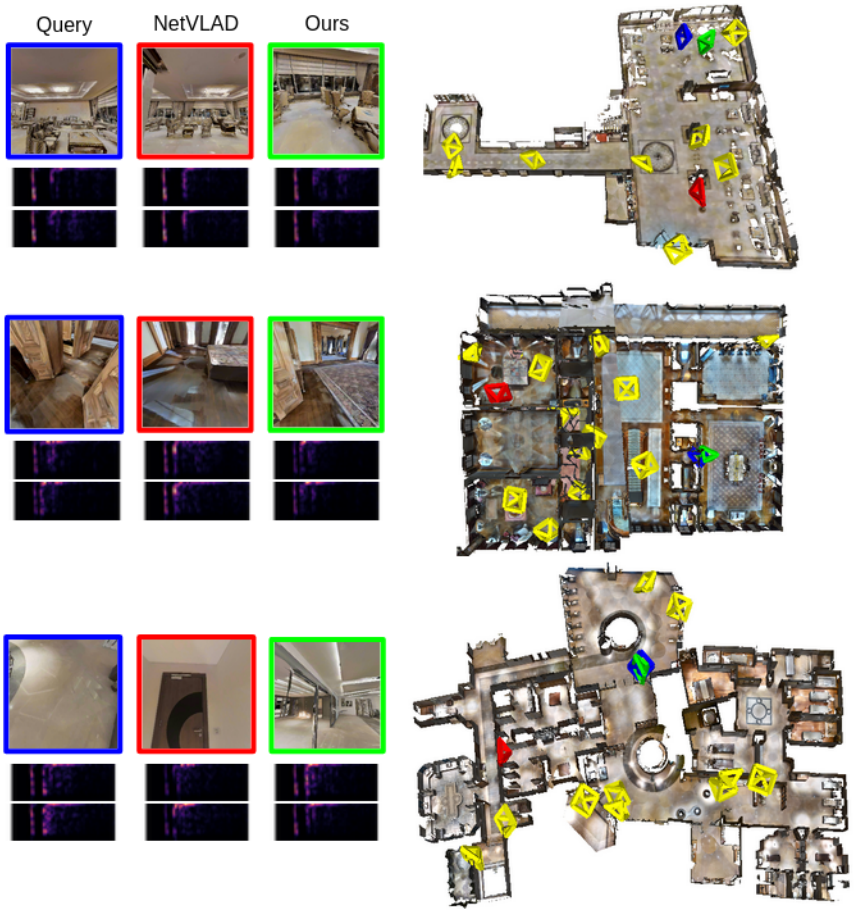

Fig. 2. Combining audio and vision for place recognition. Blue – query. Red – visually retrieved result. Green – our retrieved result. See text for details. Note that frustum rotations do not matter for accuracy, only their position.

B.7 Absolute Pose Regression: Qualitative Results

Figure 3 shows additional qualitative examples of how audio sensing benefits the vision model for absolute pose regression. The input images observe ambiguous views of the scene (blue). This result in poor performance on the part of the visual model (red), whereas our audio-visual model uses audio to disambiguate the position of the device (green).

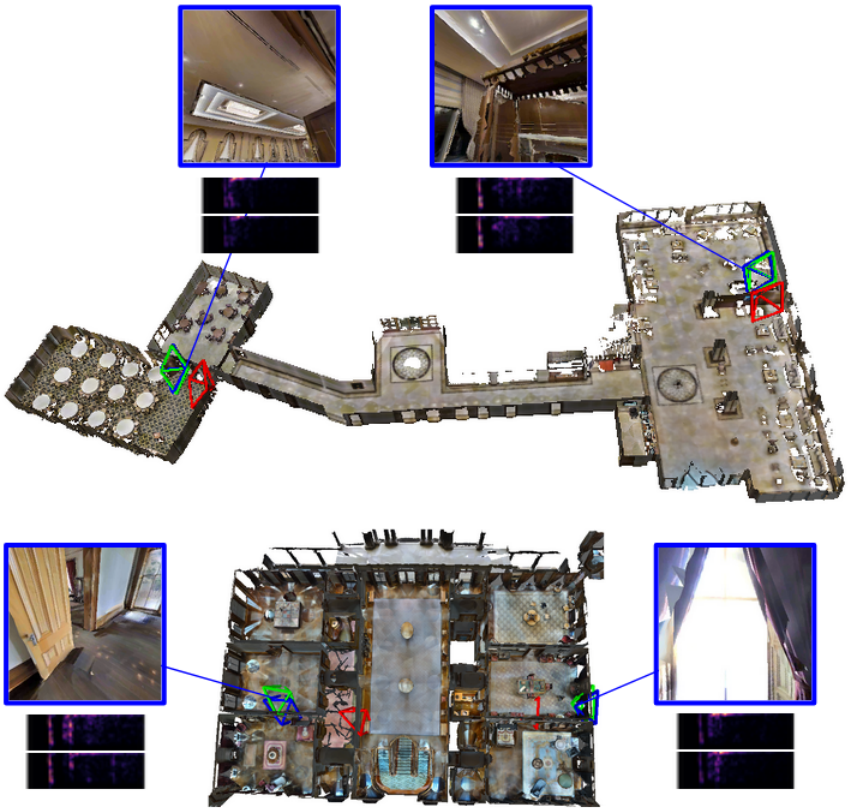

Fig. 3. Combining audio and vision for absolute pose regression. Blue – input images and frustums. Red – visual prediction. Green – our audio-visual prediction. See text for details.

References

1. Arandjelovic, R., Gronat, P., Torii, A., Pajdla, T., Sivic, J.: NetVLAD: CNN architecture for weakly supervised place recognition. In: CVPR (2016)
2. Balntas, V., Li, S., Prisacariu, V.: RelocNet: Continuous Metric Learning Relocalisation using Neural Nets. In: The European Conference on Computer Vision (ECCV) (September 2018)
3. Chang, A., Dai, A., Funkhouser, T., Halber, M., Niessner, M., Savva, M., Song, S., Zeng, A., Zhang, Y.: Matterport3D: Learning from RGB-D data in indoor environments. In: 3DV (2017)
4. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
5. Eaton, J., Gaubitch, N.D., Moore, A.H., Naylor, P.A.: The ACE challenge — corpus description and performance evaluation. In: 2015 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA) (2015)
6. Gao, R., Chen, C., Al-Halah, Z., Schissler, C., Grauman, K.: VisualEchoes: Spatial image representation learning through echolocation. In: ECCV (2020)
7. Gordo, A., Almazan, J., Revaud, J., Larlus, D.: End-to-end learning of deep visual representations for image retrieval. IJCV (2017)
8. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
9. Humenberger, M., Cabon, Y., Guerin, N., Morat, J., Revaud, J., Rerole, P., Pion, N., de Souza, C., Leroy, V., Csurka, G.: Robust Image Retrieval-based Visual Localization using Kapture. arXiv:2007.13867 (2020)
10. Kendall, A., Grimes, M., Cipolla, R.: PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization. In: ICCV (2015)
11. Laskar, Z., Melekhov, I., Kalia, S., Kannala, J.: Camera relocalization by computing pairwise relative poses using convolutional neural network. In: ICCV Workshops (2017)
12. McFee, B., Raffel, C., Liang, D., Ellis, D.P., McVicar, M., Battenberg, E., Nieto, O.: librosa: Audio and music signal analysis in python. In: Proceedings of the 14th python in science conference. vol. 8, pp. 18–25. Citeseer (2015)
13. Parida, K.K., Srivastava, S., Sharma, G.: Beyond image to depth: Improving depth prediction using echoes. In: CVPR (2021)
14. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in pytorch (2017)
15. Revaud, J., Almazan, J., Rezende, R., de Souza, C.: Learning with average precision: Training image retrieval with a listwise loss. In: ICCV (2019)
16. Sarlin, P.E., Cadena, C., Siegwart, R., Dymczyk, M.: From Coarse to Fine: Robust Hierarchical Localization at Large Scale. In: CVPR (2019)
17. Sarlin, P.E., DeTone, D., Malisiewicz, T., Rabinovich, A.: Superglue: Learning feature matching with graph neural networks. In: CVPR (2020)
18. Straub, J., Whelan, T., Ma, L., Chen, Y., Wijmans, E., Green, S., Engel, J.J., Mur-Artal, R., Ren, C., Verma, S., Clarkson, A., Yan, M., Budge, B., Yan, Y., Pan, X., Yon, J., Zou, Y., Leon, K., Carter, N., Briales, J., Gillingham, T., Mueggler, E., Pesqueira, L., Savva, M., Batra, D., Strasdat, H.M., Nardi, R.D., Goesele, M., Lovegrove, S., Newcombe, R.: The Replica dataset: A digital replica of indoor spaces. arXiv preprint arXiv:1906.05797 (2019)

19. Türkoğlu, M.Ö., Brachmann, E., Schindler, K., Brostow, G., Monszpart, A.: Visual Camera Re-Localization Using Graph Neural Networks and Relative Pose Supervision. In: 3DV. IEEE (2021)
20. Winkelbauer, D., Denninger, M., Triebel, R.: Learning to localize in new environments from synthetic training data. In: ICRA (2021)
21. Zhang, Z.: Improved adam optimizer for deep neural networks. In: 2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS). pp. 1–2. IEEE (2018)
22. Zhou, Q., Sattler, T., Pollefeys, M., Leal-Taixé, L.: To Learn or Not to Learn: Visual Localization from Essential Matrices. In: ICRA (2019)