Audio-Visual Floorplan Reconstruction Supplementary Material

Senthil Purushwalkam*[1] Sebastia Vicenc Amengual Gari[2] Vamsi Krishna Ithapu[2]
Carl Schissler[2] Philip Robinson[2] Abhinav Gupta[3] Kristen Grauman[3,4]

[1]*Carnegie Mellon University* [2]*Facebook Reality Labs* [3]*Facebook AI Research* [4]*University of Texas at Austin*

1. Additional Interior Map Visualizations

Figure 1 presents additional AV-Map interior map prediction visualizations, like Fig 3 in the main text. We see again how our model sees beyond the visible portions (cyan) to more fully map the space. We also highlight our failure modes; see the mis-classified locations (circled) on the predicted maps. We observe that the errors often arise in challenging locations that are not visually covered, where the model relies on the audio signal (see Figure 1 sample 1,4,5). Some errors arise from noise in the scan of the environment (see Figure 1 sample 3 - missing point cloud) since the rendered RGB frames are noisy.

2. Room Map Visualizations and Confusion Analysis

In Figure 2, we present additional visualizations for the estimated room maps. The room maps were generated by the AV-Map model operating in the environment-generated all-room audio setting. Green dots on the ground truth indicate the camera positions. From these visualizations, we observe that the model can successfully identify the approximate locations of several rooms. Some sources of errors are errors in interior estimation (see Column 1, Row 4 and Column 2, Row 3) and errors in localization of the rooms (see Column 1, Row 3).

In Figure 3, we present a confusion matrix for the pixel-wise room label predictions. We observe that there is a bias towards predicting the "bathroom", "hallway" and "bedroom" classes which are the three most frequent room labels. The two least frequent classes ("stairs" and "closet") are almost never predicted. This indicates that our model could benefit from training on a larger, more diverse and more balanced dataset. We also find that the rooms that are usually in close proximity have slightly higher confusion rates - for example, bedroom vs bathroom, and dining room vs kitchen. This suggests that our model struggles to accurately localize the boundaries of rooms (as also indicated in the main text).

3. Importance of Sequence Modeling

At each time step of a video, the audio clip a_i is generated by convolving a downloaded audio clip c with an impulse response ω_i. Therefore, the audio clip can be expressed as:

$$a_i = c \circledast \omega_i \tag{1}$$

OR

$$\mathbb{F}(a_i) = \mathbb{F}(c)\mathbb{F}(\omega_i) \tag{2}$$

where $\mathbb{F}$ is the Fourier transform. The impulse response encodes the acoustic characteristics of the environment for the given source location and receiver location pair at the time step i. These acoustic characteristics of the environment strongly depend on the geometric and material properties of the environments. Therefore, in order to infer the geometric properties of the environment, a model should ideally be able to either disentangle the impulse response ω_i from the audio clip a_i or infer a function of the impulse response from a_i. This is not possible from the audio clip at a single time step unless the audio clip c is known apriori. However, listening to audio clips from multiple time steps a_i, a_j, can allow us to model relative changes in impulse responses as:

$$\frac{\mathbb{F}(a_i)}{\mathbb{F}(a_j)} = \frac{\mathbb{F}(\omega_i)}{\mathbb{F}(\omega_j)} = \text{relative change in impulse response} \tag{3}$$

These relative changes can also provide information about the geometric properties of the environment. For example, walking past a door of a room containing a sound source will see a large change in impulse response clearly indicating the presence of an opening. Note here that the inferred relative change in impulse response does not rely on the

*work done while interning at Facebook AI Research. Project webpage: http://www.cs.cmu.edu/~spurushw/publication/avmap

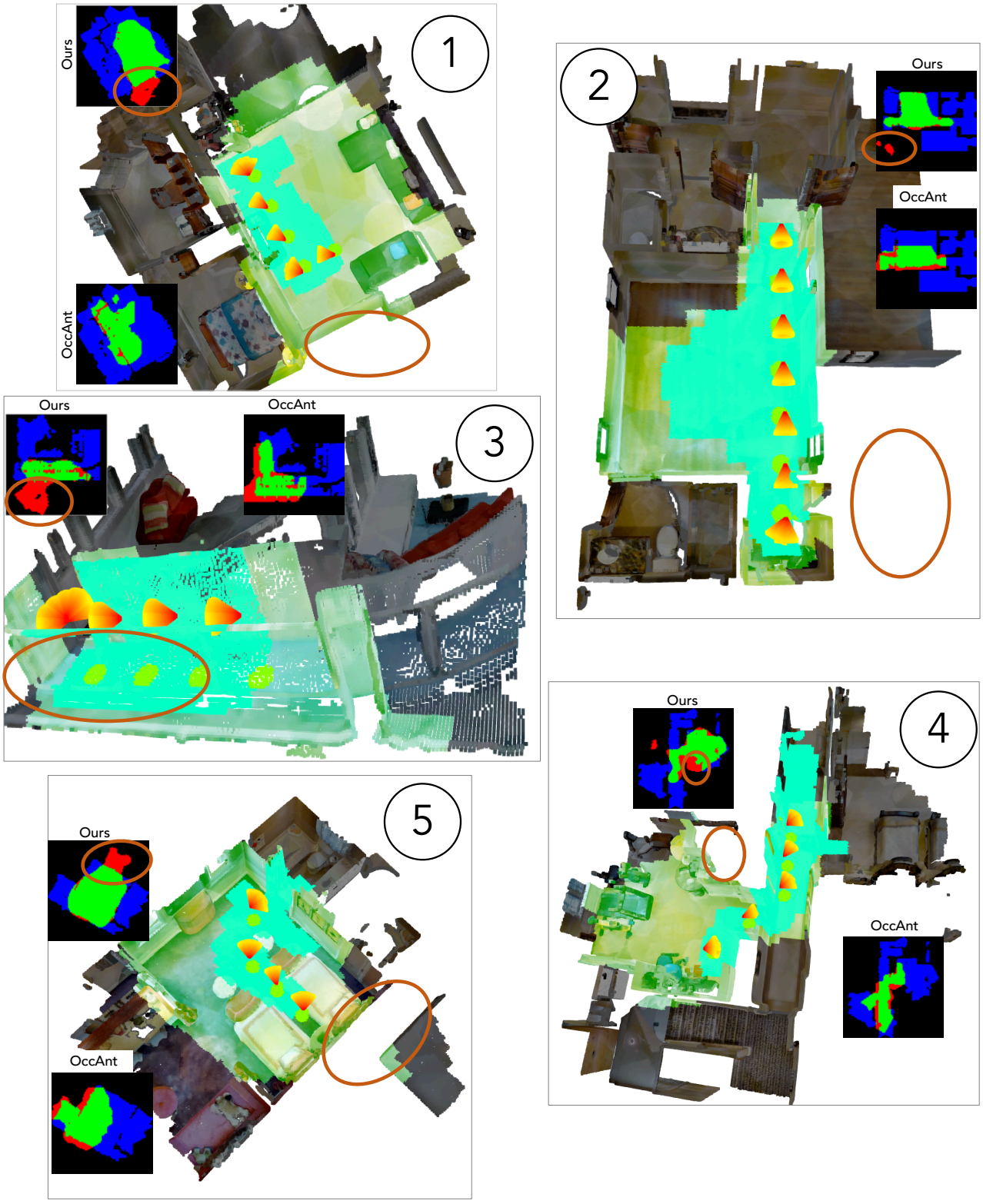


Figure 1: Additional Interior Map Visualizations: We present additional visualizations of the estimated Interior Maps. The circled areas indicate the locations that are misclassified by our proposed model. See text for discussion.

Figure 2: Additional Room Map Visualizations

original audio clip c anymore. This is also a favorable feature since downloaded audio clips are not 100% anechoic. So in practice the audio clips c encode some amount of the acoustic characteristics of the recording environment *i.e.*

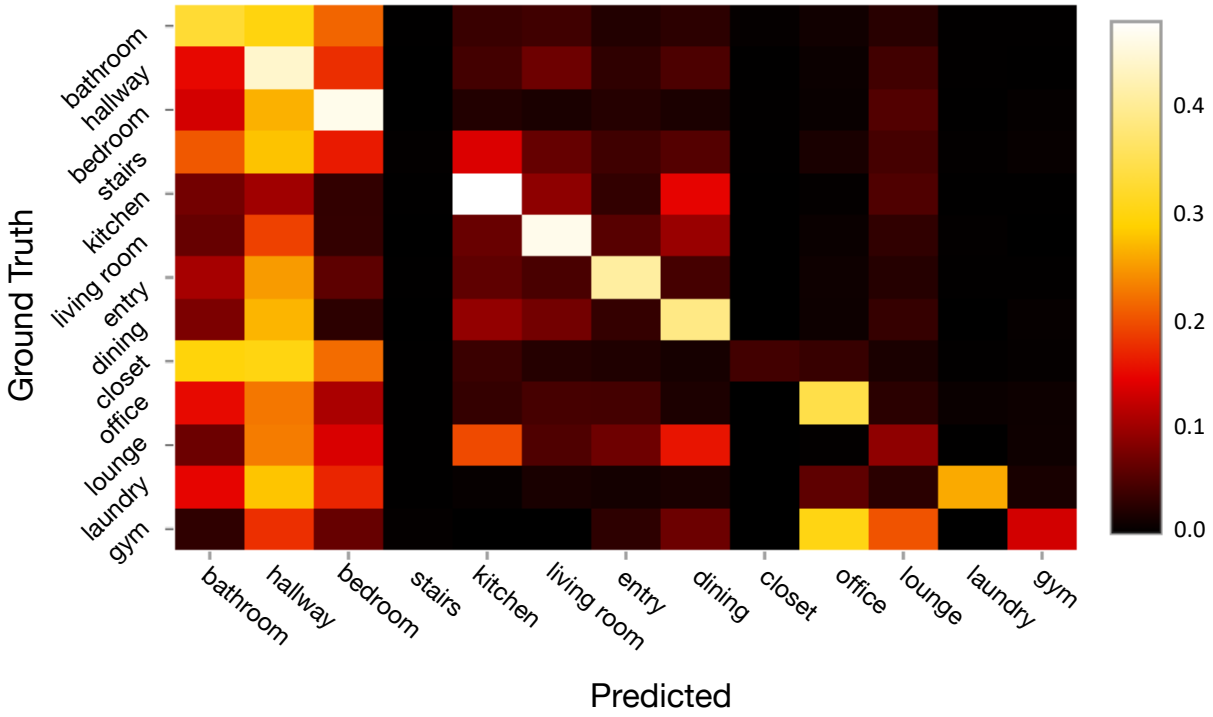


Figure 3: Room Class Confusion Matrix

$c = \hat{c} \circledast \omega_{rec}$ where $\hat{c}$ is the anechoic audio and ω_{rec} is the impulse response of the recording setup. While we do not explicitly enforce the AV-Map model to infer these relative changes, training with multiple audio clips forces the model to learn to disentangle the effect of the impulse response.

The proposed AV-Map model allows training and testing with video sequences of arbitrary length. During training, the primary bottleneck for using very long sequences is the memory footprint and speed of computation. Training with $t_V = 1$ is equivalent to making independent predictions at each time step and pooling them to obtain the final interior map estimate. For each time step, such a model would not be able to make inferences using visual features in other time steps (for example, the fact that the camera entered a door in the first step provides additional context at the second time step). Furthermore, as explained above, making independent predictions does not allow us to model relative changes in the impulse responses. We observed that sequences of length $t_V = 4$ provide the benefits of modeling sequences while maintaining a manageable training duration. As promised in the main paper, in Table 1, we show results with $t_V = 1$ and compare to the $t_V = 4$ setting to demonstrate the impact of sequence modeling. Our choice of self-attention layers is motivated by the need to model (bidirectional) temporal sequences (see Supp Sec 3) and permit variable sequence lengths. We performed an ablative experiment to verify that modeling the same sequences using LSTM layers leads to inferior performance. We observed a drop of 0.96% in accuracy, 3.57 AP, and 1.96 Edge AP in the RGB-only setting.

Table 1: Impact of Sequence Modeling: We observe a significant improvement in performance of our AV-Map model when trained on sequences compared to making independent predictions at each step.

	AP	Acc.	Edge AP
Single Step ($t_V = 1$)	68.91	62.46	54.05
Sequence ($t_V = 4$)	**73.28**	**66.52**	**54.67**

4. Predicting $164m^2$ interior area

In the main text, we presented a quantitative analysis of the AV-Map model trained to estimate interior maps for an area of $40m^2$ around the camera at each time step (by setting hyper-parameters H, W). As promised in Section 4.1 of the main paper, here in Table 2, we present similar quantitative results[1] for a model trained to predict a $164m^2$ area around the camera at each step. We observe similar results demonstrating the improved performance of the AV-Map model compared to the RGB-only model.

[1]Note that the positive and negative pixels are balanced by reweighting as discussed in Section 4.1

Table 2: Interior Map Average Precision: We present a qualitative analysis of various models trained to predict an interior area covering $164m^2$ at each time step.

Number of Steps →	1	2	4	8	16
RGB only	72.05	72.45	72.60	75.05	76.00
Dev. Gen.	**73.59**	**75.00**	**75.10**	**78.75**	**80.71**
Env. Nearby	73.01	73.36	74.08	78.03	79.76
Env. All Room	72.33	73.55	74.76	77.67	80.29

5. Effect of Varying Prediction Area

The proposed AV-Map model estimates the interior and room maps around the camera up to a certain distance at each step. Here, we analyze the effect of training AV-Map models with different prediction areas. We compare to the baseline RGB-only variant of our model and show that the full AV-Map model consistently outperforms the RGB-only model. See Table 3.

Table 3: Effect of Prediction Area: We compare the interior map average precision of AV-Map variants trained to predict different areas at each step.

	10m^2	**40**m^2	**90**m^2
RGB-only	71.58	71.07	65.86
Dev. Gen.	76.72	73.28	71.82

6. Role of Vision and Audio in AV-Map

In order to develop a deeper understanding of our proposed AV-Map model, we study the dependence of the model on each modality of data *i.e.* RGB frames and audio clips. We consider the full AV-Map model operating in the device generated audio setting. For this model, we compute the gradients of the RGB feature map and audio feature maps (output of the top-down alignment step) w.r.t the output estimated interior maps. We do this using the Guided Back-propagation approach proposed in [3]. For each gradient map, we compute the maximum value. In Figure 4, we show a scatter plot of the pair of these maximum gradient values for the two modalities for all the test samples. We observe that on average the model is significantly more sensitive to the changes in audio compared to RGB - as indicated by the larger magnitude of gradients.

In Figure 5, we present a visualization of the gradient maps for the RGB and audio features. Recall that the camera start location is at the center of these maps. It is evident that the RGB features primarily focus close to the center *i.e.* near the camera. However, the focus of audio features is more distributed, often with heavier attention to areas that extends farther than the focus of RGB features.

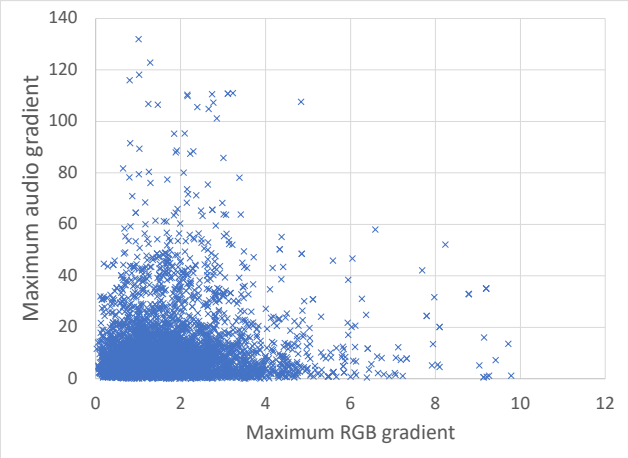


Figure 4: Impact of Audio: Here we show a scatter plot of the (maximum gradient of the RGB features w.r.t output, maximum gradient of the audio features w.r.t output) pairs.

7. Additional Dataset Details

Environments

We use the Matterport3D [1] dataset to generate video sequences (see Sec 3.4 of the main text). We use the splits provided by the SoundSpaces [2] dataset for training, validation, and testing. We include the environments in the splits here for reference:

Train environments: ['17DRP5sb8fy', '1LXtFkjw3qL', '1pXnuDYAj8r', '29hnd4uzFmX', '5LpN3gDmAk7', '5q7pvUzZiYa', '759xd9YjKW5', '7y3sRwLe3Va', '82sE5b5pLXE', '8WUmhLawc2A', 'aayBHfsNo7d', 'ac26ZMwG7aT', 'B6ByNegPMKs', 'b8cTxDM8gDG', 'cV4RVeZvu5T', 'D7N2EKCX4Sj', 'e9zR4mvMWw7', 'EDJbREhghzL', 'GdvgFV5R1Z5', 'gTV8FGcVJC9', 'HxpKQynjfin', 'i5noydFURQK', 'JeFG25nYj2p', 'JF19kD82Mey', 'jh4fc5c5qoQ', 'kEZ7cmS4wCh', 'mJXqzFtmKg4', 'p5wJjkQkbXX', 'Pm6F8kyY3z2', 'pRbA3pwrgk9', 'PuKPg4mmafe', 'PX4nDJXEHrG', 'qoiz87JEwZ2', 'rPc6DW4iMge', 's8pcmisQ38h', 'S9hNv5qa7GM', 'sKLMLpTHeUy', 'SN83YJsR3w2', 'sT4fr6TAbpF', 'ULsKaCPVFJR', 'uNb9QFRL6hY', 'Uxmj2M2itWa', 'V2XKFyX4ASd', 'VFuaQ6m2Qom', 'VVfe2KiqLaN', 'Vvot9Ly1tCj', 'vyrNrziPKCB', 'VzqfbhrpDEA', 'XcA2TqTSSAj', 'D7G3Y4RVNrH', 'E9uDoFAP3SH', 'JmbYfDe2QKZ', 'r1Q1Z4BcV1o', 'r47D5H71a5s', 'ur6pFq6Qu1A', 'VLzqgDo317F', 'YmJkqBEsHnH', 'ZMojNkEp431']
Val environments: ['2azQ1b91cZZ', '8194nk5LbLH', 'EU6Fwq7SyZv', 'oLBMNvg9in8', 'QUCTc6BB5sX', 'TbHJrupSAjP', 'X7HyMhZNoso', 'pLe4wQe7qrG', 'x8F5xyUWy9e', 'Z6MFQCViBuw', 'zsNo4HB9uLZ']
Test environments: ['5ZKStnWn8Zo', 'ARNzJeq3xxb',

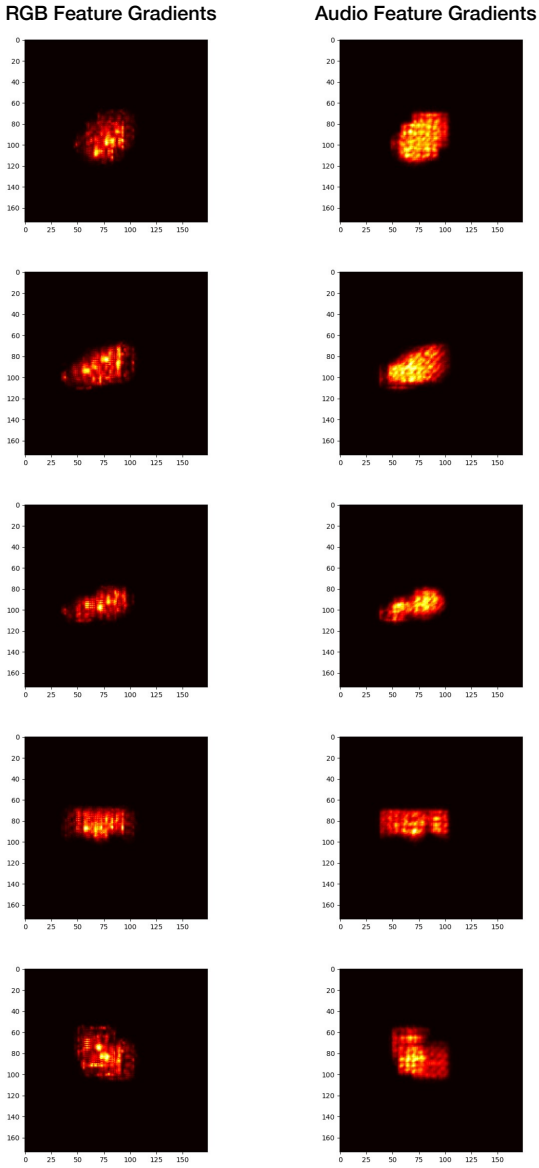


Figure 5: Complementary Strengths of Audio and RGB: We present the gradient maps of RGB and Audio features show that the vision helps with prediction in regions near the camera (center) while audio has a more distributed focus. We also observe that the attention of audio features reaches beyond the limits of RGB features' focus.

'fzynW3qQPVF', 'jtcxE69GiFV', 'pa4otMbVnkk', 'q9vSo1VnCiC', 'rqfALeAoiTq', 'UwV83HsGsw3', 'wc2JMjhGNzB', 'WYY7iVyf5p8', 'YFuZgdQ5vWj', 'yqstnuAEVhm', 'gxdoqLR6rwA', 'gYvKGZ5eRqb', 'Vt2qJdWjCF2']

Room types and associated sounds

For generating room maps, we choose the 13 most frequent room types. For each room type, we download sounds from www.freesound.org generated by objects (or people) that are unique to the room type. Here we present the list of rooms, their associated sounds, and the number of train/val/test sounds for each:

- bathroom: brushing (3/2/2), flush(4/1/1)
- hallway: <no sound>
- bedroom: alarm clock (5/3/3)
- stairs: footsteps (5/3/3)
- kitchen: blender (3/1/1), cabinet (1/1/1), dishwasher (3/2/2)
- living room: telephone (5/3/3)
- entryway/foyer/lobby: knock (5/2/2)
- dining room: knife (4/1/1) , spoon(4/2/2)
- closet: closet door (2/2/2)
- office: keyboard (5/3/3)
- lounge: no sound
- laundryroom/mudroom: washing machine (5/3/3)
- workout/gym/exercise: person panting (5/3/3)

8. Implementation and Training Details

8.1. Hyperparameters

The AV-Map model is trained with a batchsize of 32 videos using 4 GPUs. Each sample in the batch is generated by randomly sampling a camera trajectory as described in the main text. We use the SGD optimizer with a starting learning rate of 0.1, momentum 0.9 and weight decay 0.00001. After 30000 SGD updates, we drop the learning rate to 0.01 and train for an additional 20000 SGD steps.

8.2. Positional Encoding

The positional encoding map added in the feature alignment stage (see Sec 3.2) is a 64-channel 2D map representing the position of each pixel with a 64 dimensional vector. For position (i,j) in the feature map, the positional encoding $PE(i,j)$ is computed as:

$$PE(i) = \Big[\sin(\frac{i}{10000^{0/32}}), \cos(\frac{i}{10000^{0/32}}), \sin(\frac{i}{10000^{2/32}}), \cos(\frac{i}{10000^{2/32}}), \ldots, \sin(\frac{i}{10000^{30/32}}), \cos(\frac{i}{10000^{30/32}})\Big] \tag{4}$$

$$PE(i,j) = [PE(i), PE(j)] \tag{5}$$

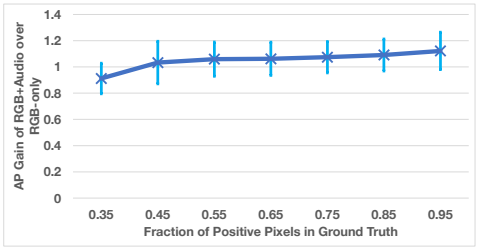
8.3. Feature Alignment

Here we present a pseudo-code to illustrate the feature alignment described in Section 3.2.

```
# f_t: visual features at time-step t
# g_t: audio features at time-step t
# r_t=(x_t, y_t, θ_t): relative position in meters and angle
# res: per-pixel feature resolution (of f_t,g_t) in meters

max_feat_dim = max(f_t.shape[-1],f_t.shape[-2])
max_disp = max([x_1/res, x_2/res, ..., x_t_v/res, y_1/res, y_2/res, ..., y_t_v/res])

# Calculate amount to pad
# = maximum displacement.+ sqrt(2)* feature dimension
# 2nd term accounts for rotation by 45°
padding = max_disp + max_feat_dim*sqrt(2)


# def concat_position_embedding: append 64 position embedding channels (see Sec F.2)
# def pad(f,n): expand each spatial dimension by 2n (n before and n after) and fill with zeros
# def translate(f, (x,y)): move the feature vertically by y pixels and horizontally by x pixels
# def rotate(f, θ): rotate feature by θ about the center of the feature map


for t in 1,2,...,t_V:
   f_t = concat_position_embedding(f_t)
   f'_t = pad(f_t,padding)
   f'_t = translate(f'_t, (x_t/res, y_t/res))
   f'_t = rotate(f'_t, θ_t)


for t in 1,2,...,t_V:
   g_t = concat_position_embedding(g_t)
   g'_t = pad(g_t,padding)
   g'_t = translate(g'_t, (x_t/res, y_t/res))
   g'_t = rotate(g'_t, θ_t)
```

9. Interior Positive-Negative Imbalance

In order to disentangle the effect of positive/negative imbalance, we balance their contribution in the evaluation metrics (as explained in Sec 4.1). Here we visualize the trend of relative gain in Interior Prediction AP of our full AV-Map model over the RGB-only variant w.r.t the fraction of positive pixels in the test samples. Here we note that the errors are almost uniform across different ratios of positive pixels. Hence our results are not artificially boosted by the balance of occupied/freespace.



References

[1] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *arXiv preprint arXiv:1709.06158*, 2017. 5

[2] Changan Chen, Unnat Jain, Carl Schissler, Sebastia Vicenc Amengual Gari, Ziad Al-Halah, Vamsi Krishna Ithapu, Philip Robinson, and Kristen Grauman. SoundSpaces: Audio-visual navigation in 3d environments. In *ECCV*, 2020. 5

[3] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014. 5