



MOTIVATION

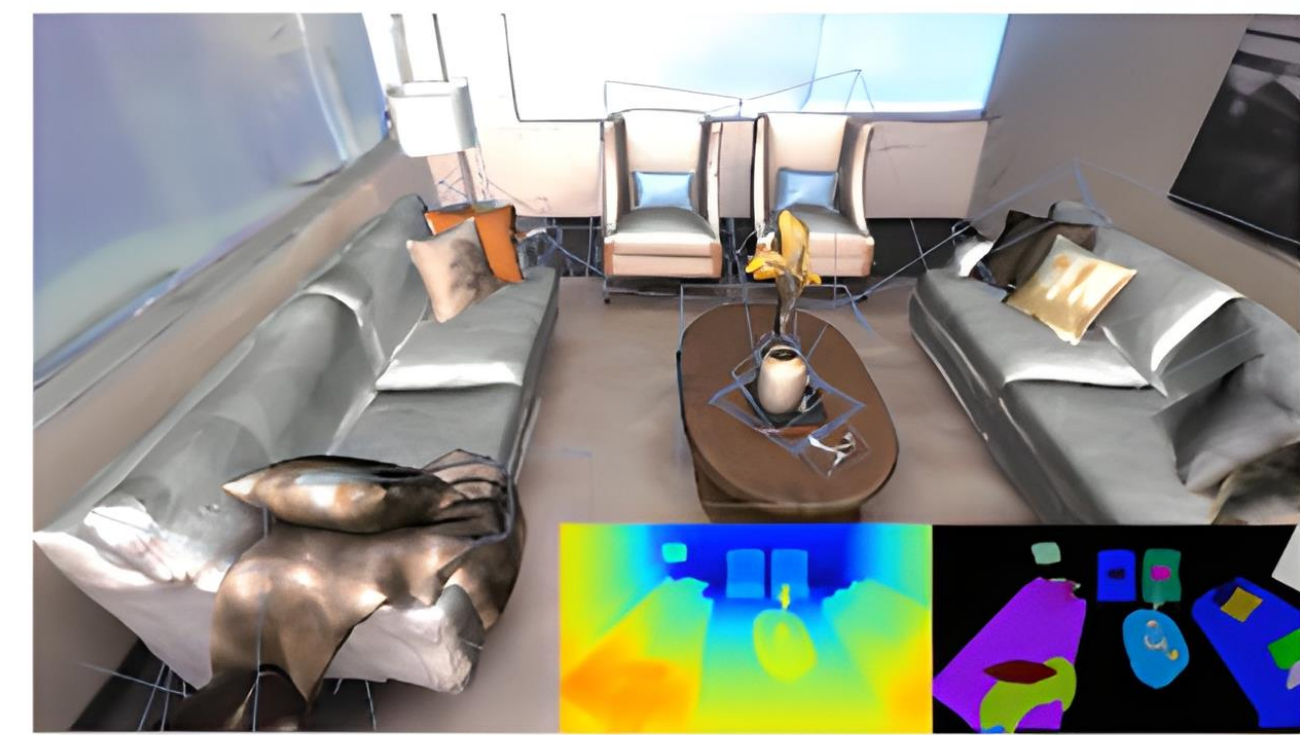
Input: An RGB-D sequence

Output: Per-object implicit model parameterised by tiny MLPs

- ✓ Object-level Neural Field SLAM
- ✓ One MLP (Neural Implicit) per Object
- ✓ Real-time Vectorised Multi-Network Training

Neural Field SLAM	<i>iMAP (ICCV'21)</i> <i>NICE-SLAM (CVPR'22)</i>	<ul style="list-style-type: none"> No Decomposition Unknown Correspondences
Decomposed NeRF	<i>KiloNeRF (ICCV'21)</i> <i>BlockNeRF (CVPR'22)</i>	<ul style="list-style-type: none"> Split in 3D Space Not Semantically Meaningful
Semantic NeRF	<i>Object-NeRF (ICCV'21)</i> <i>FeatureField (NeurIPS'22)</i>	<ul style="list-style-type: none"> Offline Training No Independent Model
Object-level SLAM	<i>MID-Fusion (ICRA'19)</i> <i>NodeSLAM (3DV'20)</i>	<ul style="list-style-type: none"> Unoptimisable / TSDF-based Need Category Level Prior

Tracked View Reconstruction



Fixed View Reconstruction

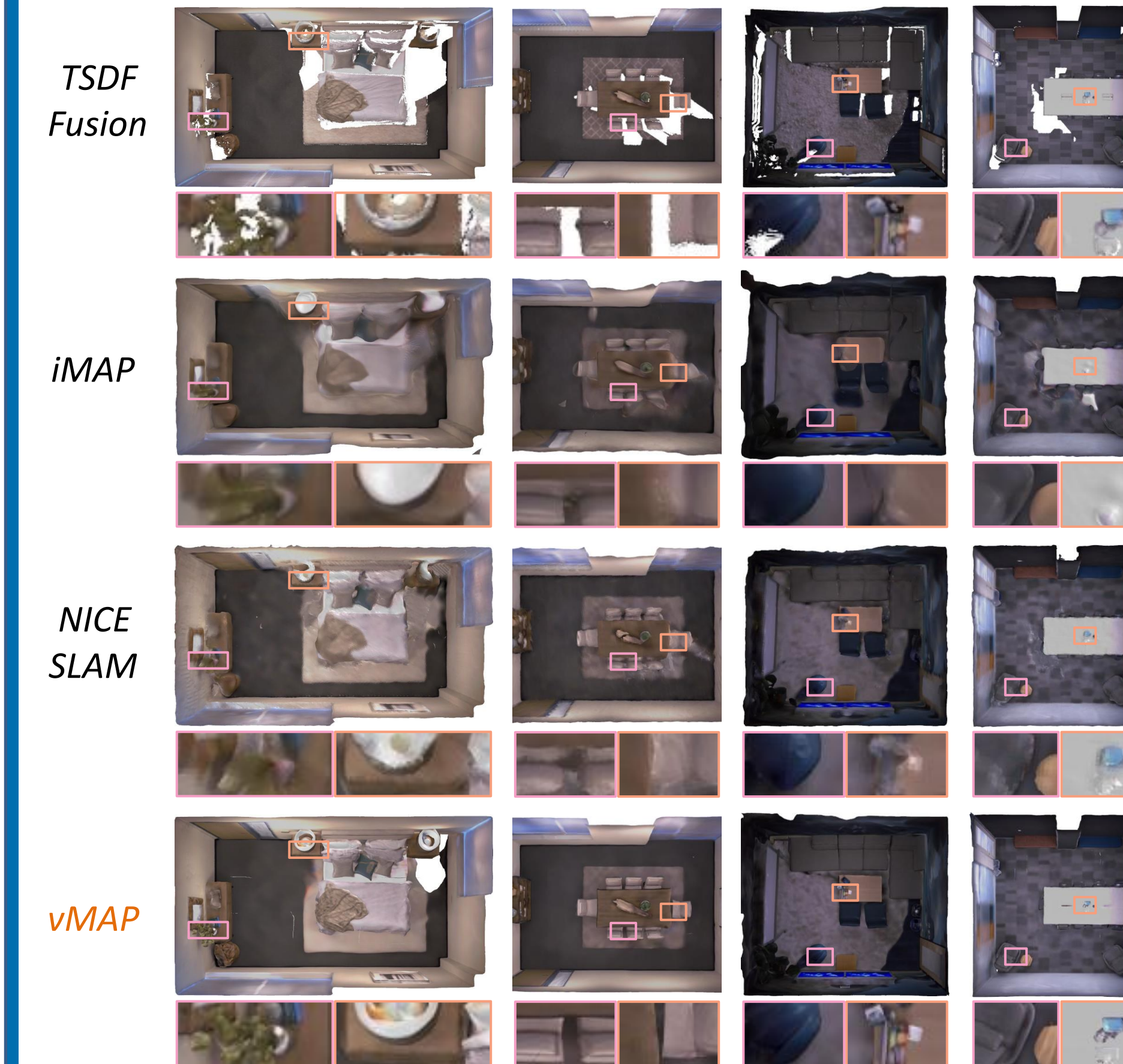


Object Reconstruction

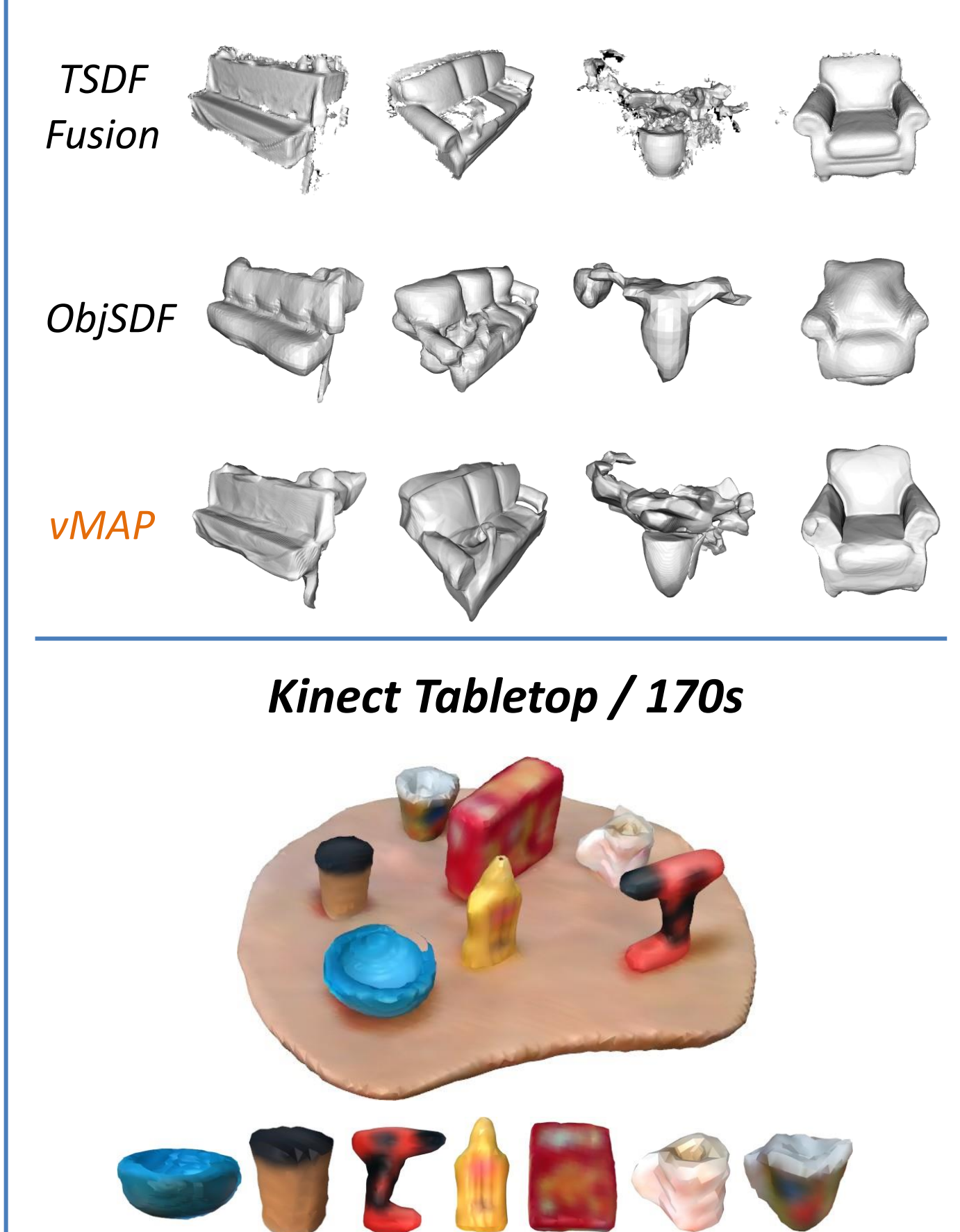


RESULTS

Replica Scenes / 6mins



ScanNet Scenes / 20 mins

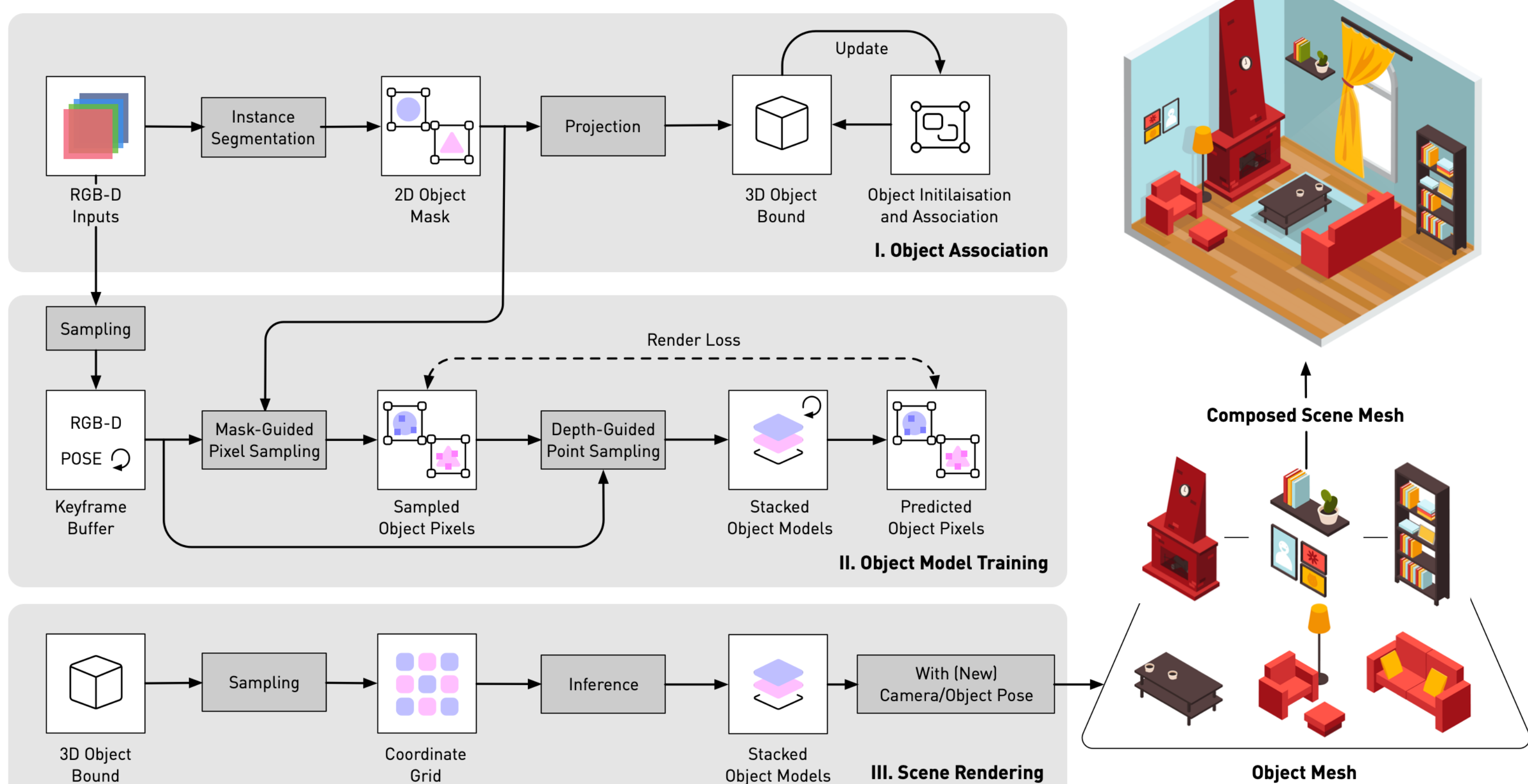


Kinect Tabletop / 170s



PIPELINE

vMAP — A Real-time Object-level Dense Neural Mapping System



VECTORISED TRAINING DETAILS

PyTorch-Level Implementation using Functorch

1. Init batch of models with the same structure.

```
fmodel, params, buffers = combine_state_for_ensemble(models)
[p.requires_grad_() for p in params]
optimiser.add_param_group({"params": params})
```

2. Get batch predictions in one go.

```
batch_pred = vmap(fmodel)(params, buffers, batch_input)
```

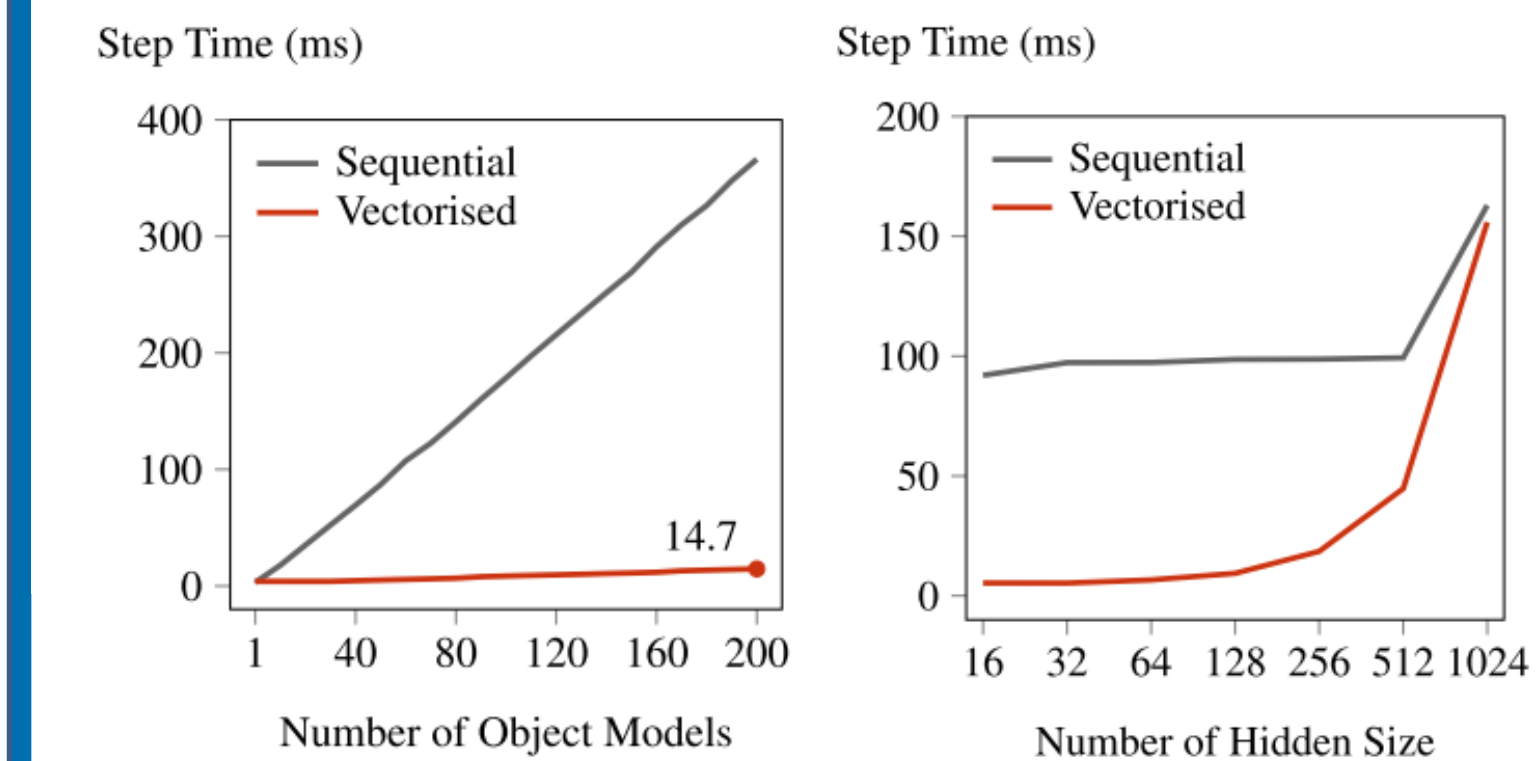
3. Back propagation.

```
batch_loss = loss(batch_pred, batch_gt)
batch_loss.backward()
optimiser.step()
optimiser.zero_grad(set_to_none=True)
```

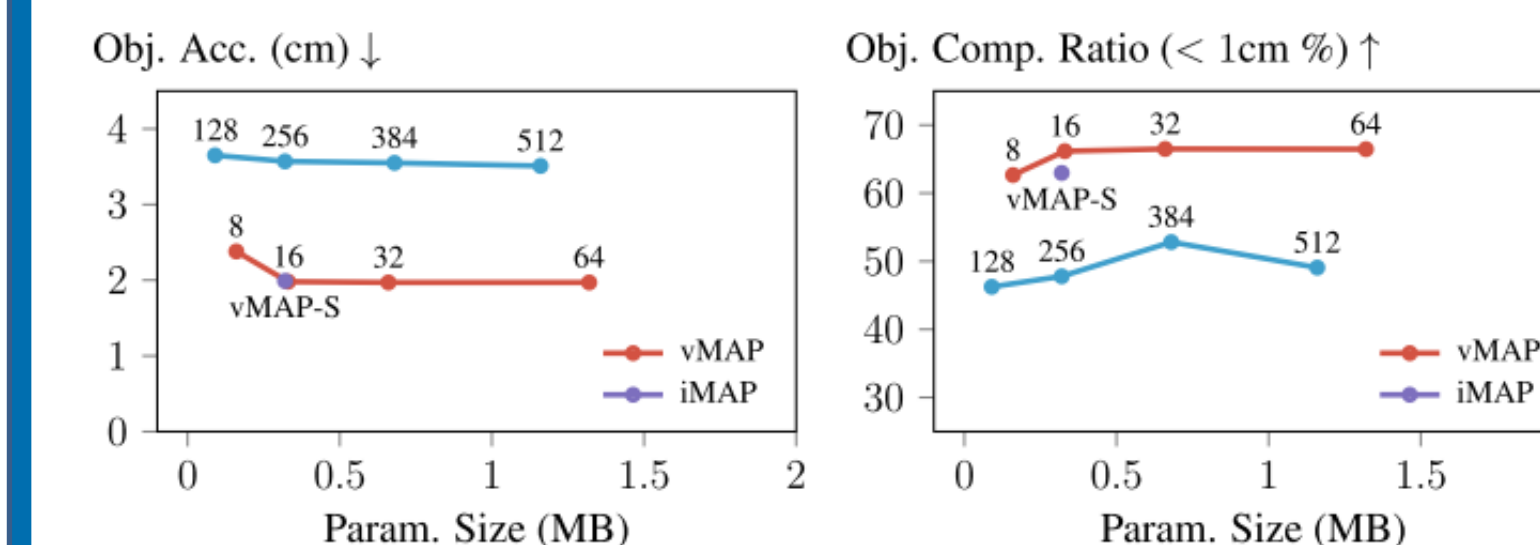
4. Update original model parameters.

```
with torch.no_grad():
    for idx, model in enumerate(models):
        for i, param in enumerate(model.parameters()):
            param.set_(params[i][idx])
```

Vectorised v.s Sequential



Quality v.s Model Size



Object Plausible Completion



Future Direction

Bring object-level web-scale priors into 3D?