TT-NF: Tensor Train Neural Fields

Anton Obukhov[1], Mikhail Usvyatsov[1], Christos Sakaridis[1], Konrad Schindler[1], Luc Van Gool[1,2]
[1]ETH Zurich [2]KU Leuven

TT-NF is a low-rank representation of a neural field defined on a multi-dimensional grid.

1. Fully-differentible sparsity;
2. Logarithmic parameters scaling;
3. Training by backprop through samples;
4. Efficient batch sampling algorithms for DL.

Where can it be used?

1. Tensor regression setting, in place of SVD-based schemes, such as TT-SVD, TT-Cross. Values y_i of the function $F(i), i=(i_1, ..., i_D)$ approximated by the D-dimensional field F_θ can be directly used to learn parameters θ of the field. See **Tensor Denoising** experiments;
2. Deep setting, in which values of F cannot be evaluated. Field parameters θ can be learned through implicit supervision of function $G_\gamma\left(F_\theta(i^{(1)}), ..., F_\theta(i^{(K)})\right)$ with parameters γ, where the choice of indices i is governed by an external process. See **NeRF** experiments.

Recap of Tensor Train Formats

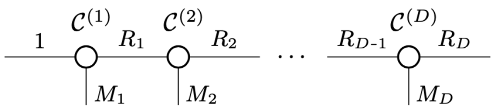
Tensor Train (TT) decomposition of a low-rank tensor of shape $M_1 \times ... \times M_D$ and rank $R = (R_0, R_1, ..., R_{D\text{-}1}, R_D)$, where $R_0 = R_D = 1$, is a product of D factors of shape $R_{i\text{-}1} \times M_i \times R_i$:

$$A_{i_1,...,i_D} = \sum_{\beta_1,...,\beta_{D\text{-}1}=1}^{R_1,...,R_{D\text{-}1}} \mathcal{C}^{(1)}_{1,i_1,\beta_1} \cdot \mathcal{C}^{(2)}_{\beta_1,i_2,\beta_2} \cdots \mathcal{C}^{(D)}_{\beta_{D\text{-}1},i_D,:}$$

Block TT is an extension for vector-valued data of shape $M_1 \times ... \times M_D \times R_D$, where $R_D > 1$.

Tensor Denoising

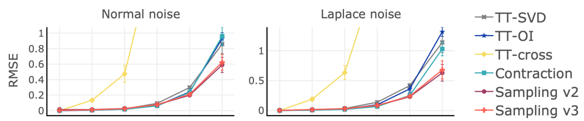
We choose TT-rank, parameterize the desired TT decomposition factors, and train it as a model with regression loss using the deep learning tooling.



Compared to prior art, TT-NF is flexibile in choice of sampling strategy and presense of noise in y.

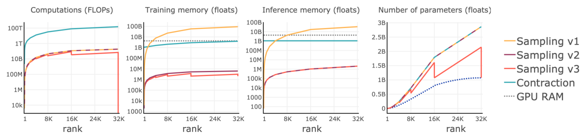
Method	Observation access pattern	Noise in observations
TT-SVD	Full tensor	Not supported
TT-Cross	On-demand, pattern defined by dimensions and TT-rank	Not supported
TT-OI	Full tensor	Sub-gaussian
TT-NF (our)	On-demand, flexible batch size and access pattern	Any supported by the choice of the loss function

TT-NF achieves best results on the synthetic task.



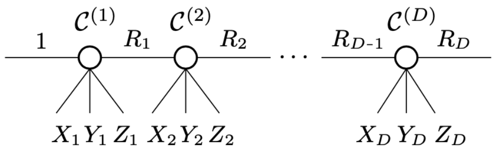
Efficient Sampling

Multi-linearity of the TT format allows multiple sampling schemes. The proposed algorithm is the only one that unlocks learning intractable volumes.



Neural Radiance Fields (NeRF)

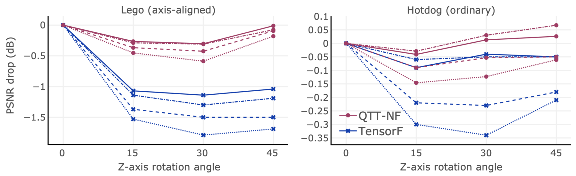
Radiance field of size $256^3 \times 28$ is parameterized using octet QTT format with 8 levels of hierarchy. The format is inspired by octrees used in graphics; allows learning large hierarchical voxel grids.



TT-NF archieves competitive results with baseline NeRF and other forms of differentiable sparsity.

Metric	Method (shading)	Chair	Drums	Ficus	Hotdog	Lego	Materials	Mic	Ship	Avg.
PSNR ↑	NeRF	33.00	25.01	30.13	36.18	32.54	29.62	32.91	28.65	31.01
	TensoRF (-mask)	32.19	25.01	30.81	35.28	33.54	28.81	31.72	28.90	30.78
	QTT-NF (SH)	32.09	24.96	30.89	35.49	32.48	28.22	31.50	27.55	30.40
	QTT-NF (NN)	32.87	25.30	31.85	35.97	33.00	28.67	33.07	27.97	31.09

Ablation of capacity invariance to content rotation.



TLDR: fully-differentiable type of sparsity for learning hierarchical multi-dimensional data or embeddings (fields) using standard deep learning tools.

Twitter:
`@antonobukhov1`
Project:
`obukhov.ai/ttnf`

ETH zürich