

u^b

^b
**UNIVERSITÄT
BERN**

AEC
ALBERT EINSTEIN CENTER
FOR FUNDAMENTAL PHYSICS

FSD LUT Siren - Update

Dominik Wermelinger

Siren Model and Loss function

Siren Model:

$$\mathbf{x} = (x, y, z) \longrightarrow \underbrace{\mathbf{h}_1 = \sin[\omega_0 (W_1 \mathbf{x} + \mathbf{b}_1)]}_{\text{first layer}} \longrightarrow \underbrace{\mathbf{h}_i = \sin[\omega_0 (W_i \mathbf{h}_{i-1} + \mathbf{b}_i)]}_{\text{hidden layers}} \longrightarrow \begin{cases} \hat{\mathbf{y}} = \sin[\omega_0 (W_{\text{out}} \mathbf{h}_L + \mathbf{b}_{\text{out}})], & \text{linear=False} \\ \hat{\mathbf{y}} = W_{\text{out}} \mathbf{h}_L + \mathbf{b}_{\text{out}}, & \text{linear=True} \end{cases}$$

MSE Loss function

$$L = \frac{1}{NC} \sum_{i=1}^N \sum_{j=1}^C w_{ij} (\hat{y}_{ij} - y_{ij})^2$$

- So far only used the linear=False due to the visibility transformations

Visibility transformations and Weights

Visibility transformation

$$t' = \begin{cases} \frac{\log_{10}(v_{raw} + \epsilon) - \log_{10}(\epsilon)}{\log_{10}(v_{max} + \epsilon) - \log_{10}(\epsilon)} \\ \text{asinh}(v/\epsilon) / \text{asinh}(v_{max}/\epsilon) \end{cases}, \quad \in [0, 1]$$

$$t = \begin{cases} t', & \text{sine=False} \quad \in [0, 1] \\ 2t' - 1, & \text{sine=True} \quad \in [-1, 1] \end{cases}$$

Occurrence Weights:

$$w_i = \frac{N_{tot}}{N_{occ} n_i},$$

Original Weights:

$$w_{ij} = \begin{cases} 10^6 \cdot v_{ij}, & \text{if } 10^6 \cdot v_{ij} \geq 10^{-8}, \\ 1, & \text{if } 10^6 \cdot v_{ij} < 10^{-8} \end{cases}$$

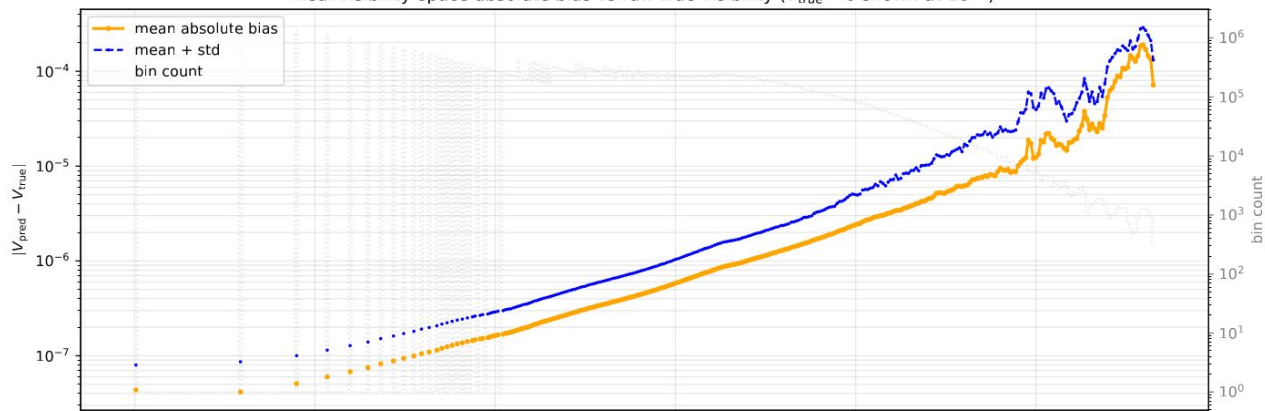
$$L = \frac{1}{NC} \sum_{i=1}^N \sum_{j=1}^C w_{ij} (\hat{t}_{ij} - t_{ij})^2$$

- MSE uses transformed visibilities to train against, and weights related to the “real-space” visibilities
- **sine=True** transformations give singularity in transformed space for mean absolute relative error

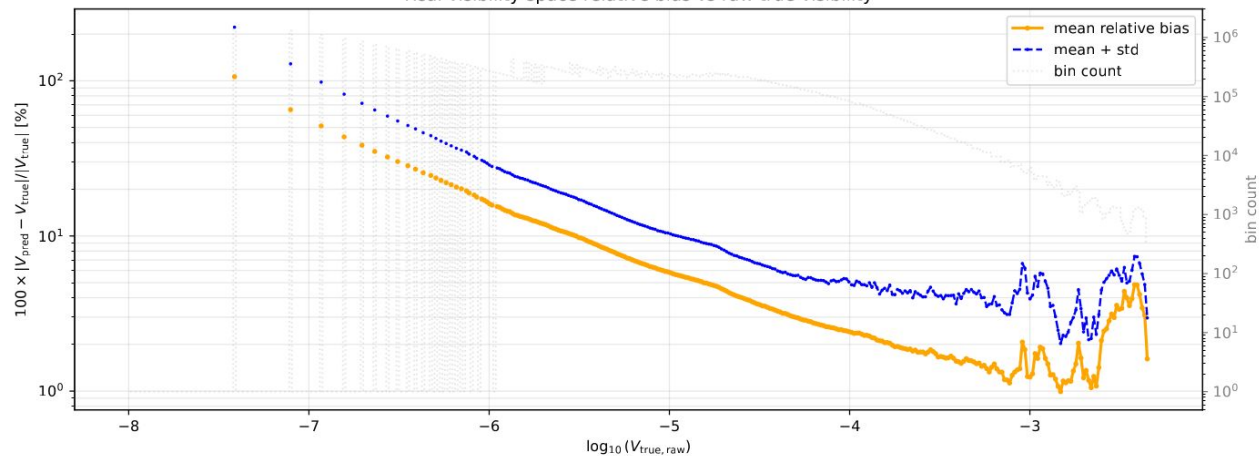
Real visibility biases (sine=True, occ weights)

Real-visibility-space binned bias vs raw true visibility (400 bins, all channels)

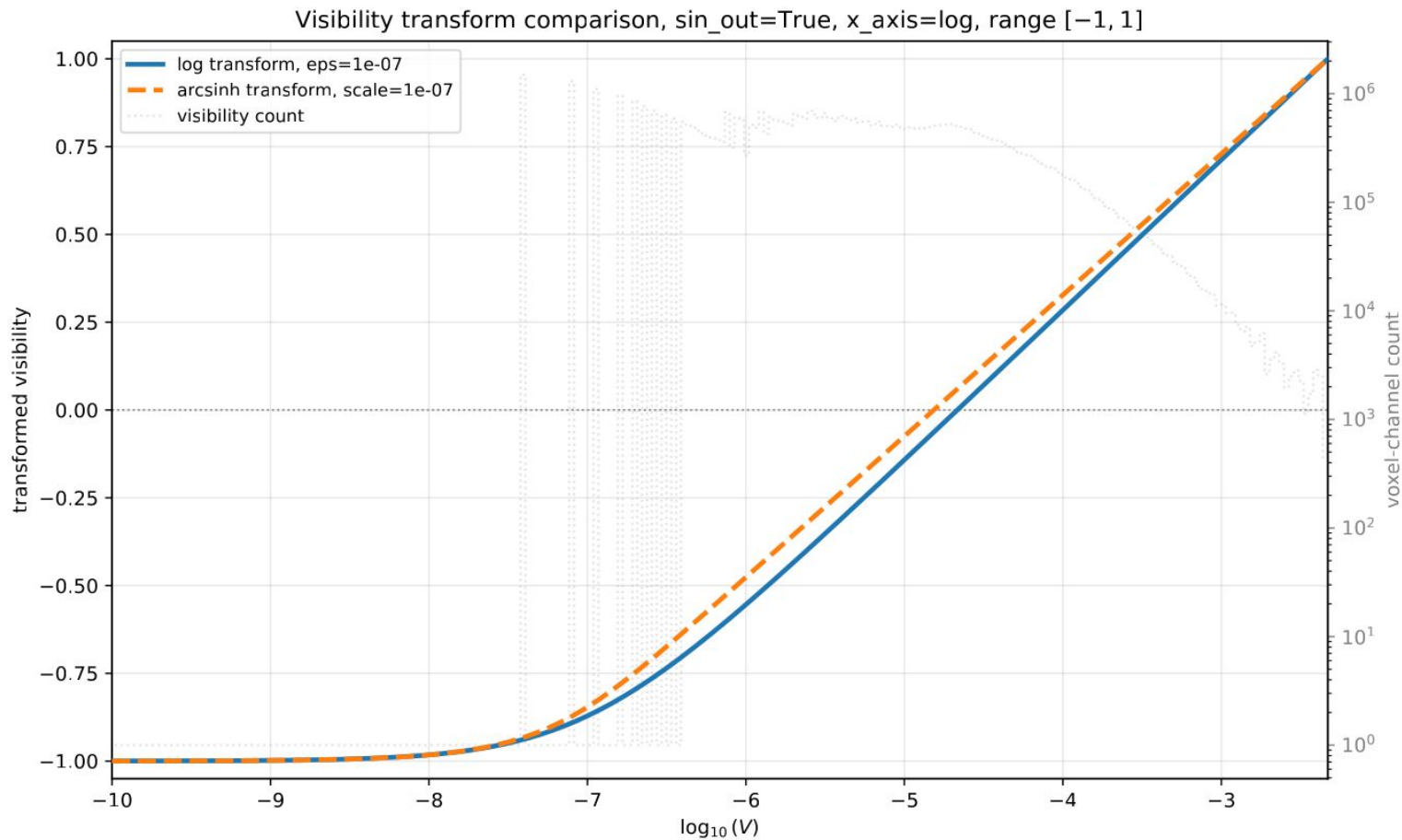
Real-visibility-space absolute bias vs raw true visibility ($V_{true} = 0$ shown at 10^{-8})



Real-visibility-space relative bias vs raw true visibility



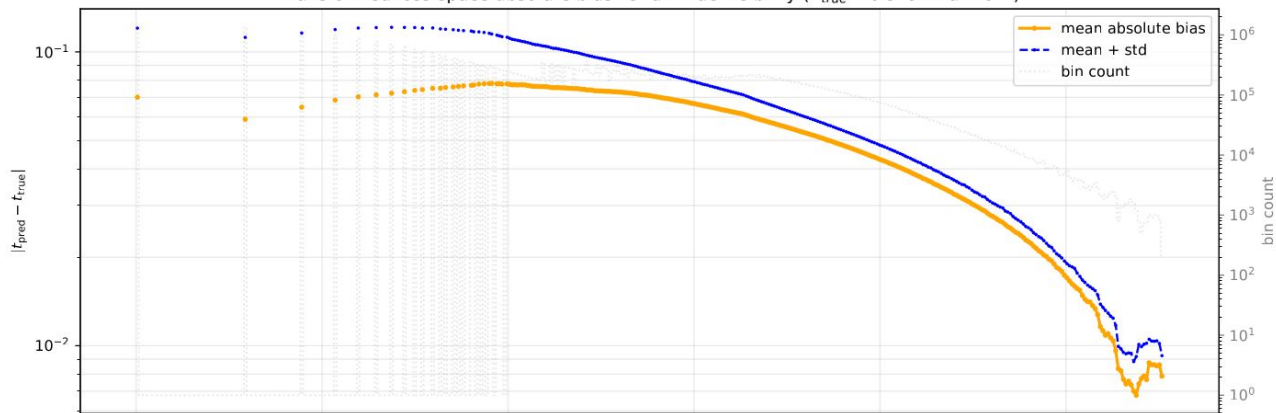
arcsinh vs log transformations



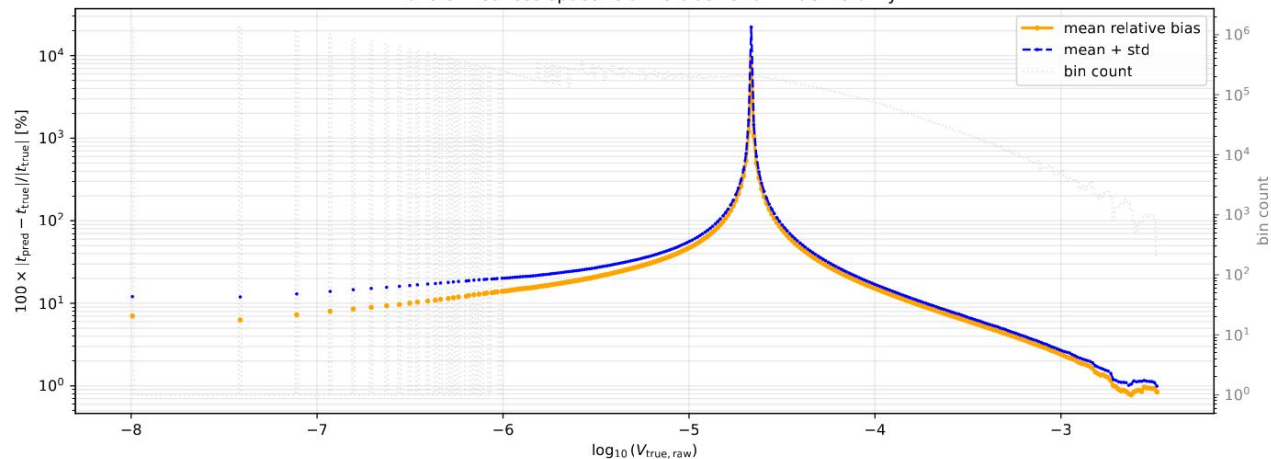
Transformed visibility biases (sine=True, occ weights)

Transformed loss-space binned bias vs raw true visibility (400 bins, all channels)

Transformed loss-space absolute bias vs raw true visibility ($V_{true} = 0$ shown at 10^{-8})



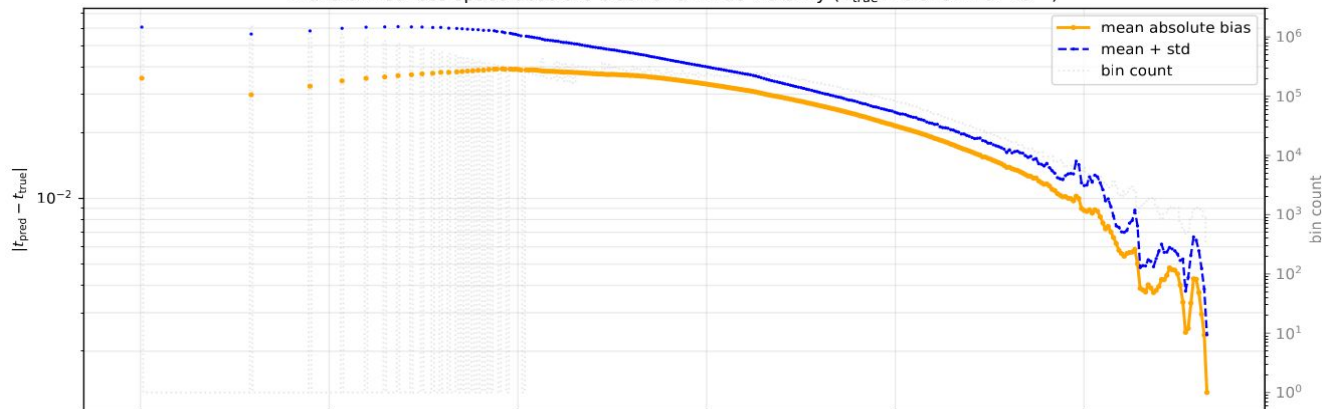
Transformed loss-space relative bias vs raw true visibility



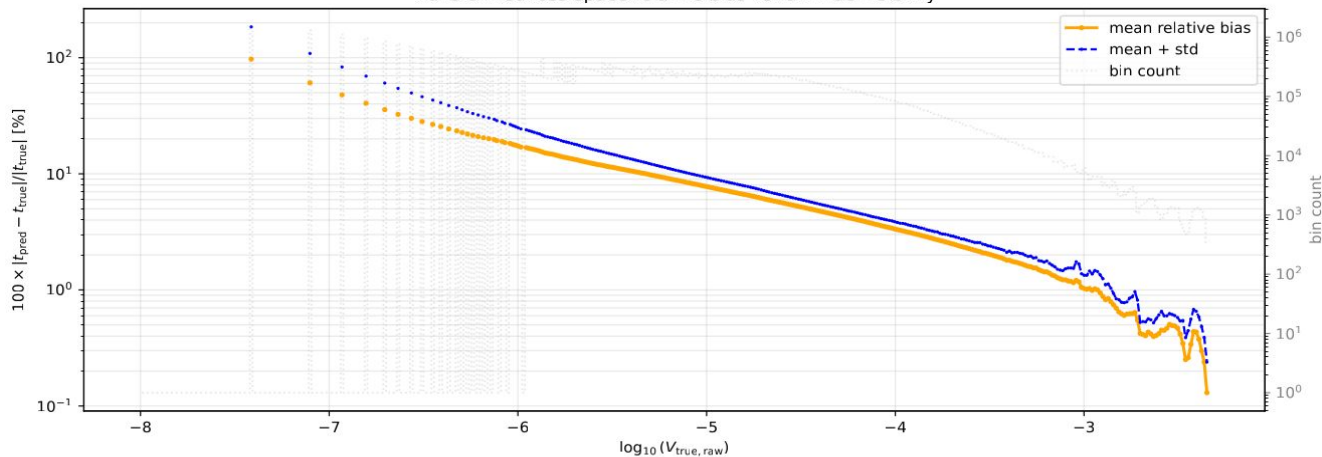
Transformed visibility biases (sine=False, occ weights)

Transformed loss-space binned bias vs raw true visibility (400 bins, all channels)

Transformed loss-space absolute bias vs raw true visibility ($V_{true} = 0$ shown at 10^{-8})



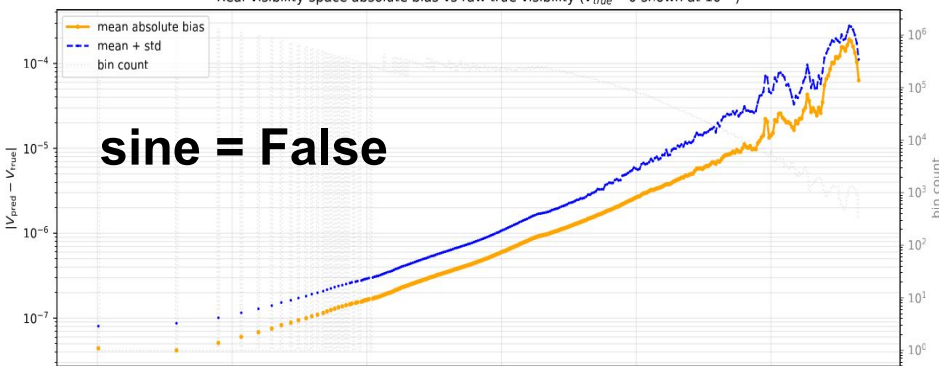
Transformed loss-space relative bias vs raw true visibility



Comparison real space biases

Real-visibility-space binned bias vs raw true visibility (400 bins, all channels)

Real-visibility-space absolute bias vs raw true visibility ($V_{true} = 0$ shown at 10^{-8})

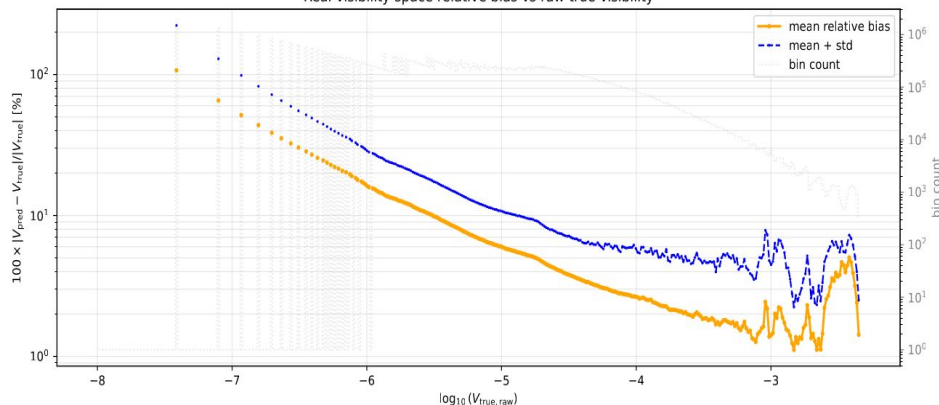


Real-visibility-space binned bias vs raw true visibility (400 bins, all channels)

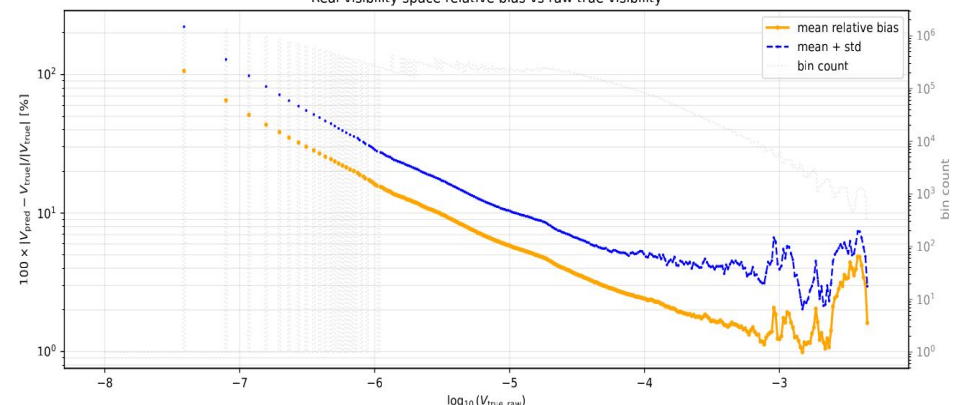
Real-visibility-space absolute bias vs raw true visibility ($V_{true} = 0$ shown at 10^{-8})



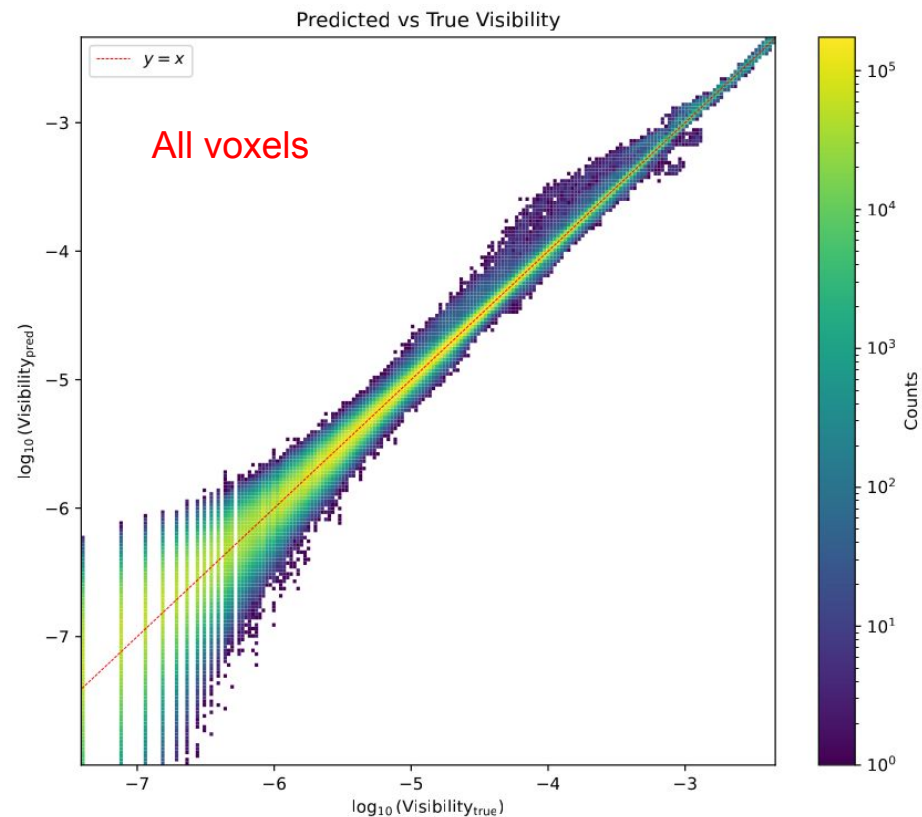
Real-visibility-space relative bias vs raw true visibility



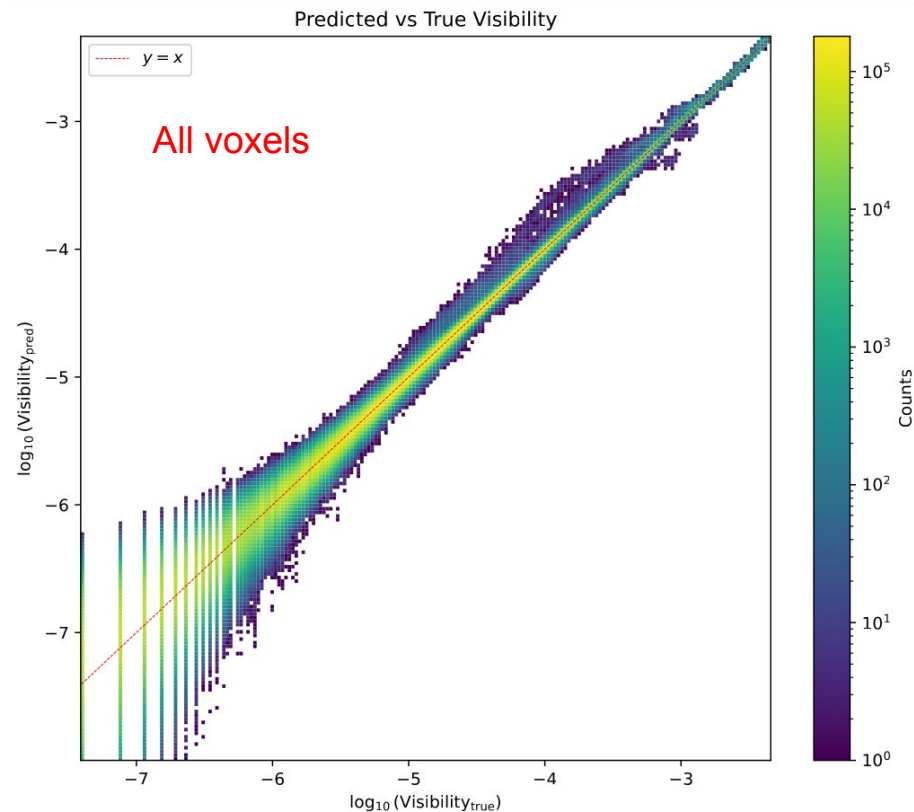
Real-visibility-space relative bias vs raw true visibility



sine = False

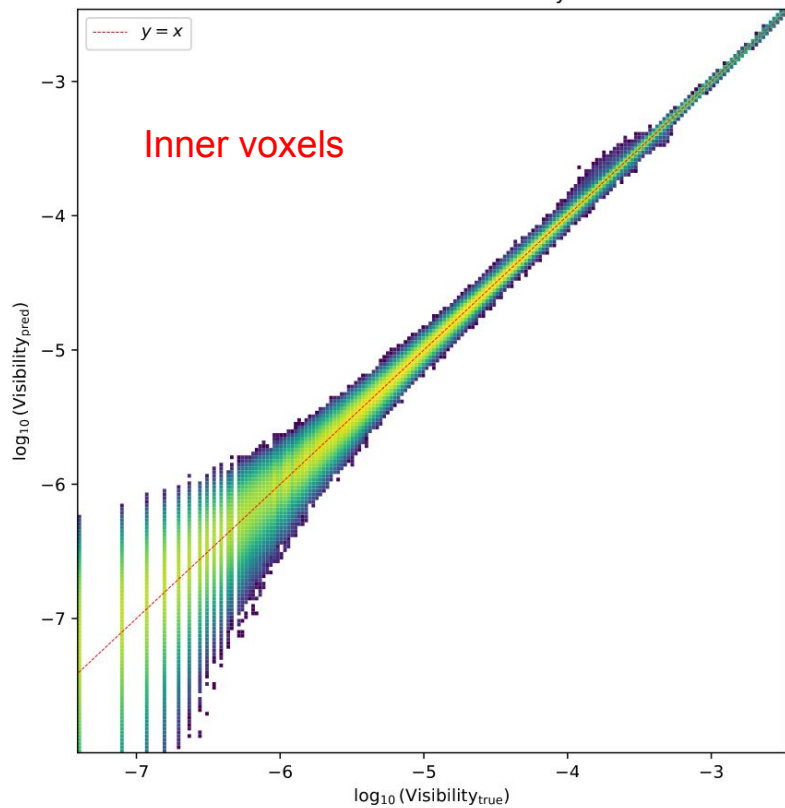


sine = True



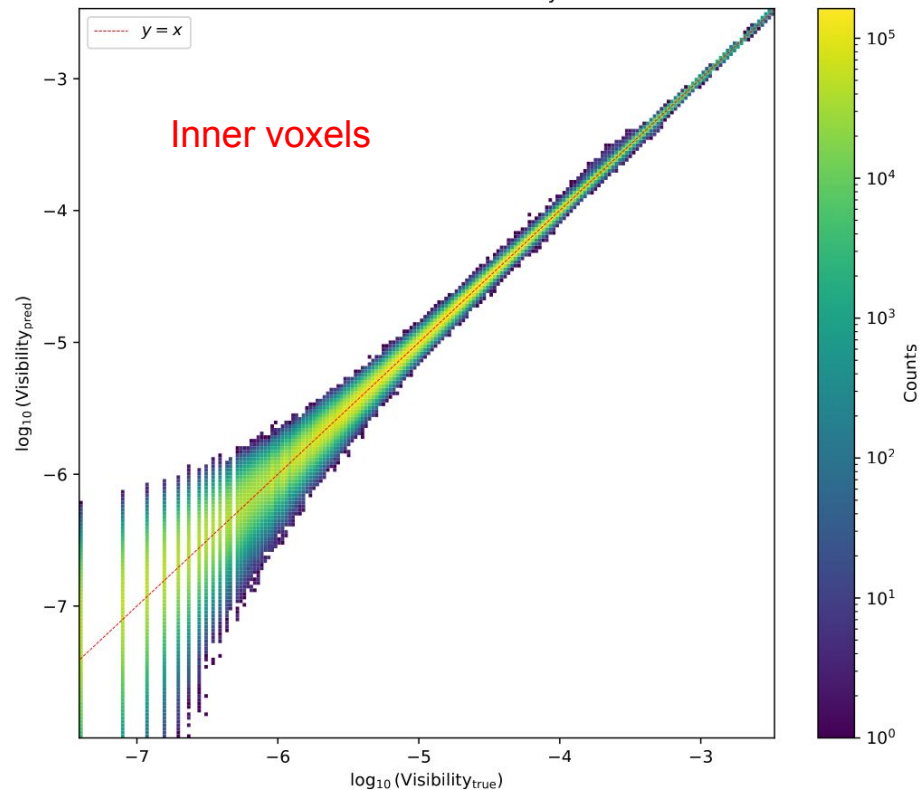
sine = False

Predicted vs True Visibility



sine = True

Predicted vs True Visibility



How to include simulation/data uncertainty

- Low visibility channels have large uncertainties
- For track data calibration, there might be channels that have dark-counts, need to account for that
- From Patrick: module 0 Siren calibration with tracks:

$$\lambda_j = Y_j \sum_{i \in \text{hits}} Q_i v_j(\mathbf{r}_i),$$

$$\chi^2 = \sum_{j \in \text{SiPMs}} \frac{(n_j - \lambda_j)^2}{\lambda_j + \epsilon_0^2}.$$

λ_j : predicted number of photoelectrons (p.e.) in SiPM j ,

Y_j : effective light yield of SiPM j [p.e./ke⁻],

Q_i : measured charge of hit i ,

$\mathbf{r}_i$: position of hit i ,

$v_j(\mathbf{r}_i)$: visibility from position $\mathbf{r}_i$ to SiPM j ,

n_j : observed light signal in SiPM j ,

ϵ_0 : light-signal uncertainty at zero signal ($\epsilon_0 = 5$ p.e.),

Uncertainty implementations for FSD LUT training

Uncertainty including loss:

$$\mathcal{L} = \frac{1}{N_v N_{\text{ch}}} \sum_{i=1}^{N_v} \sum_{c=1}^{N_{\text{ch}}} w_{i,c} \ell_{i,c}.$$

$$\ell_{i,c} = \begin{cases} z_{i,c}^2, & \chi^2 \text{ loss,} \\ [\max(0, |z_{i,c}| - 1)]^2, & \chi^2 \text{ loss with } 1\sigma \text{ tolerance.} \end{cases}$$

$$z_{i,c} = \frac{\lambda_{i,c}^{\text{model}} - k_{i,c}^{\text{LUT}}}{\sqrt{\sigma_{i,c}^2}}.$$

$$\sigma_{i,c}^2 = \begin{cases} \max(k_{i,c}^{\text{LUT}}, \sigma_0^2), & \text{variance with floor,} \\ k_{i,c}^{\text{LUT}} + \sigma_0^2, & \text{additive variance.} \end{cases}$$

- Transform model output with inverse-vis-transformation to “linear” visibilities
-> forces model to be trained in transformed space ([0,1] or [-1,1])
- Then go from linear visibilities to count-space (lambda & k)

$$\hat{y}_{i,c} = f_{\theta}(\mathbf{x}_i).$$

$$\hat{v}_{i,c} = T^{-1}(\hat{y}_{i,c}).$$

$$\lambda_{i,c}^{\text{model}} = N_{\gamma} \hat{v}_{i,c},$$

$$k_{i,c}^{\text{LUT}} = N_{\gamma} v_{i,c}^{\text{LUT}}.$$

Uncertainty implementations for FSD LUT training

Uncertainty including loss:

$$\mathcal{L} = \frac{1}{N_v N_{\text{ch}}} \sum_{i=1}^{N_v} \sum_{c=1}^{N_{\text{ch}}} w_{i,c} \ell_{i,c}.$$

weights set to 1

$$\ell_{i,c} = \begin{cases} z_{i,c}^2, & \chi^2 \text{ loss,} \\ [\max(0, |z_{i,c}| - 1)]^2, & \chi^2 \text{ loss with } 1\sigma \text{ tolerance.} \end{cases}$$

did not converge

$$z_{i,c} = \frac{\lambda_{i,c}^{\text{model}} - k_{i,c}^{\text{LUT}}}{\sqrt{\sigma_{i,c}^2}}.$$

$$\sigma_{i,c}^2 = \begin{cases} \max(k_{i,c}^{\text{LUT}}, \sigma_0^2), & \text{variance with floor,} \\ k_{i,c}^{\text{LUT}} + \sigma_0^2, & \text{additive variance.} \end{cases}$$

- Transform model output with inverse-vis-transformation to “linear” visibilities
-> forces model to be trained in transformed space [-1,1]
- Then go from linear visibilities to count-space
- Evaluate loss-function

Uncertainty implementations for FSD LUT training

Uncertainty including loss:

$$\mathcal{L} = \frac{1}{N_v N_{\text{ch}}} \sum_{i=1}^{N_v} \sum_{c=1}^{N_{\text{ch}}} w_{i,c} \ell_{i,c}.$$

weights set to 1

$$\ell_{i,c} = \begin{cases} z_{i,c}^2, & \chi^2 \text{ loss,} \\ [\max(0, |z_{i,c}| - 1)]^2, & \chi^2 \text{ loss with } 1\sigma \text{ tolerance.} \end{cases}$$

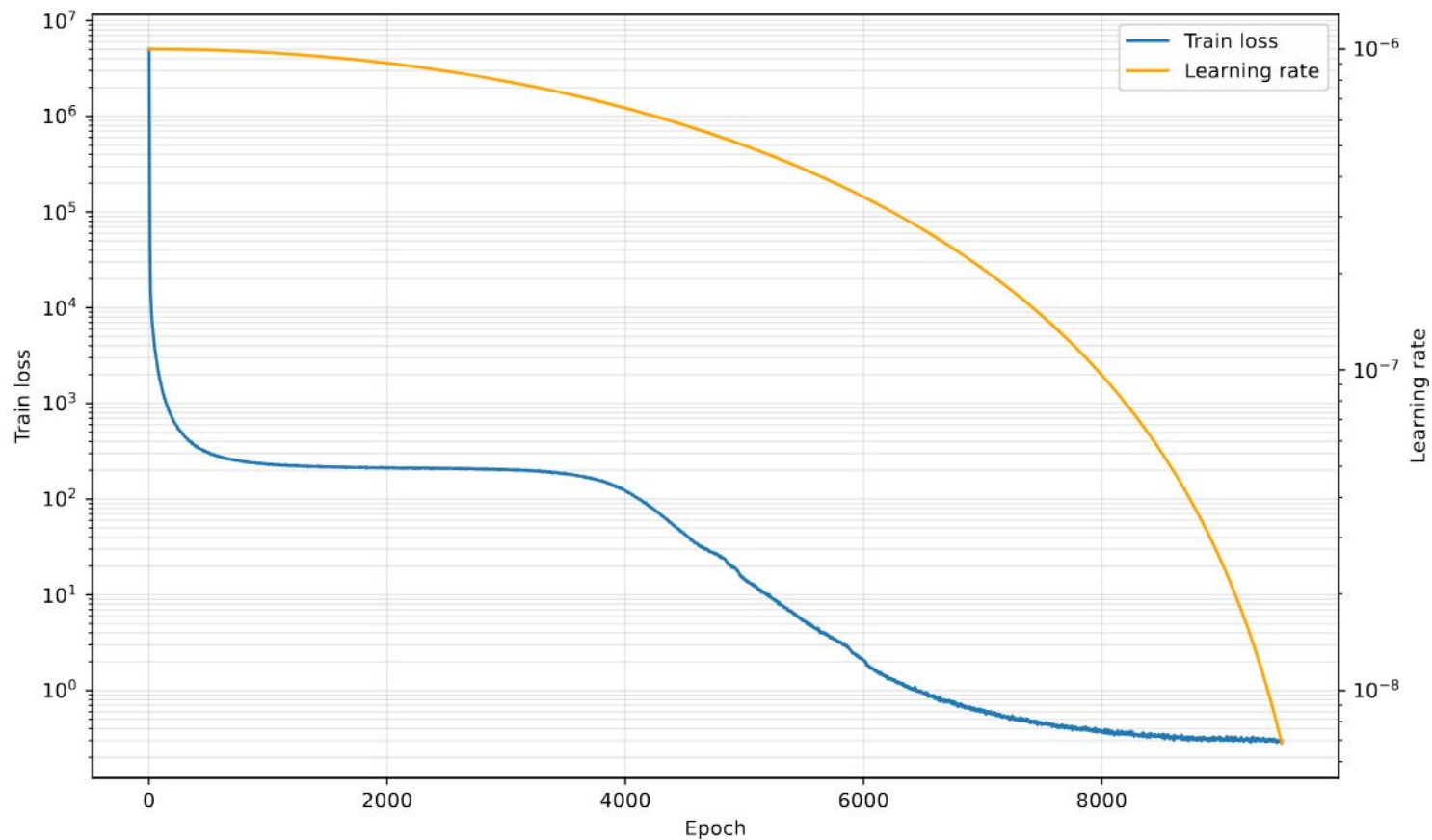
funkily converging

$$z_{i,c} = \frac{\lambda_{i,c}^{\text{model}} - k_{i,c}^{\text{LUT}}}{\sqrt{\sigma_{i,c}^2}}.$$

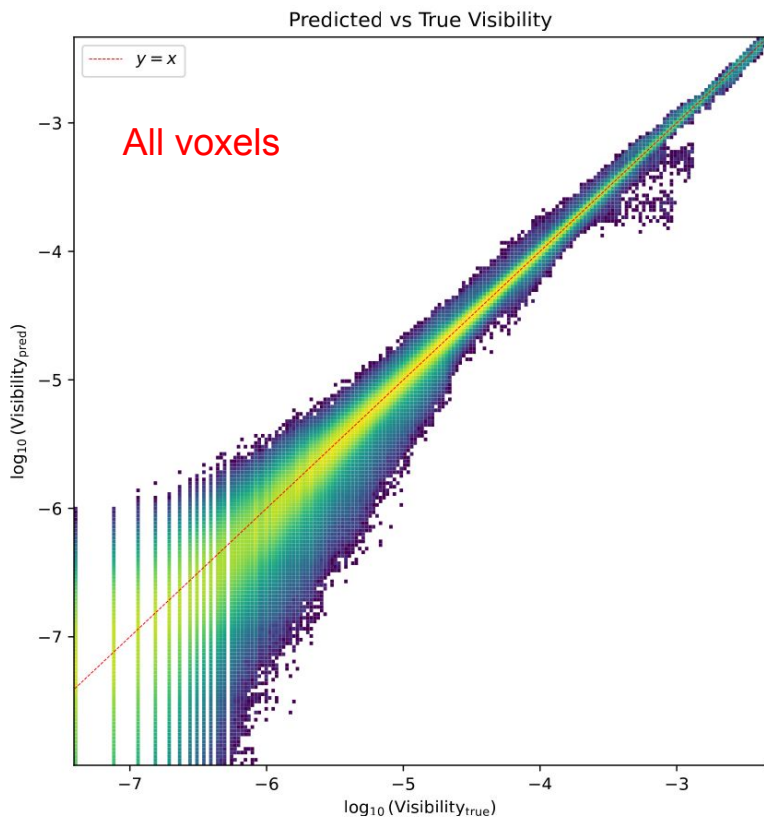
$$\sigma_{i,c}^2 = \begin{cases} \max(k_{i,c}^{\text{LUT}}, \sigma_0^2), & \text{variance with floor,} \\ k_{i,c}^{\text{LUT}} + \sigma_0^2, & \text{additive variance.} \end{cases}$$

- Transform model output with inverse-vis-transformation to “linear” visibilities
-> forces model to be trained in transformed space [-1,1]
- Then go from linear visibilities to count-space
- Evaluate loss-function

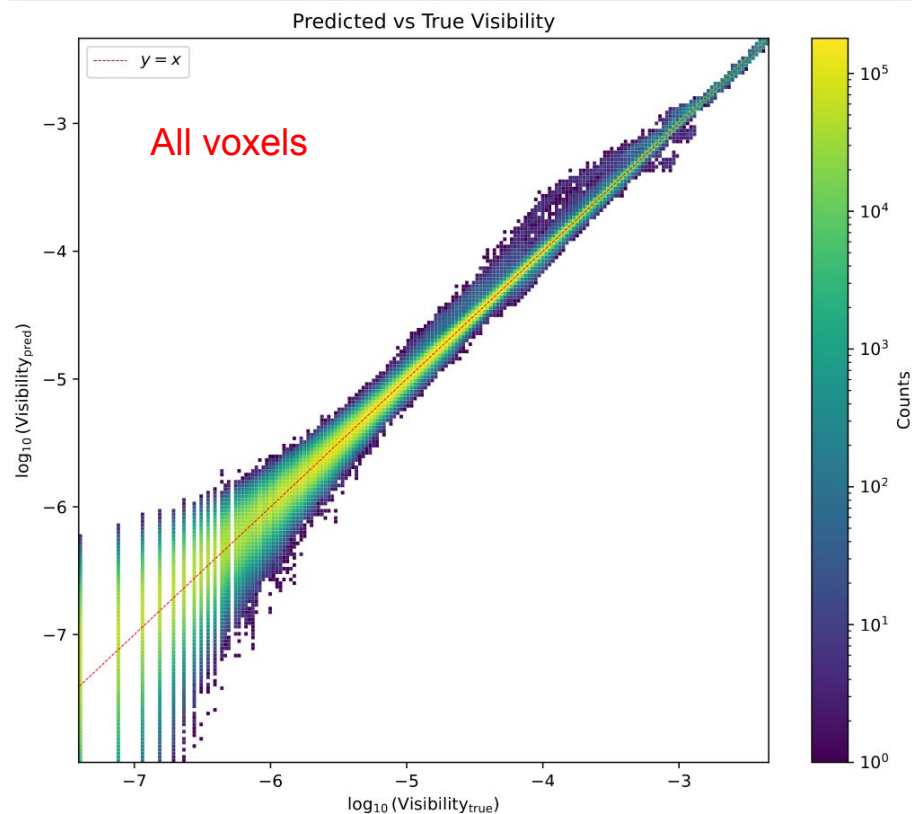
Chi² loss & learning rate



Chi² loss

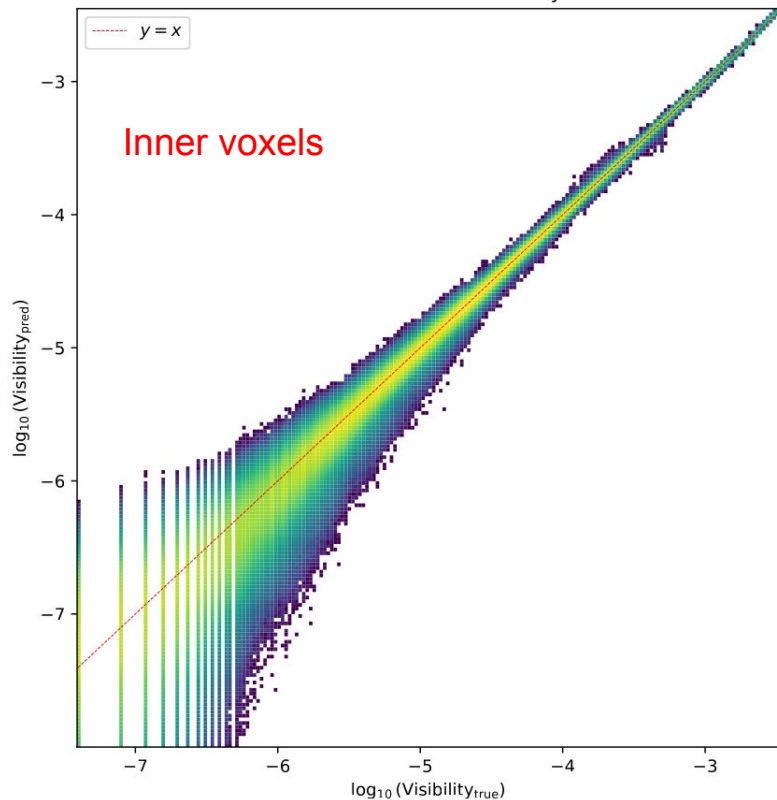


MSE loss



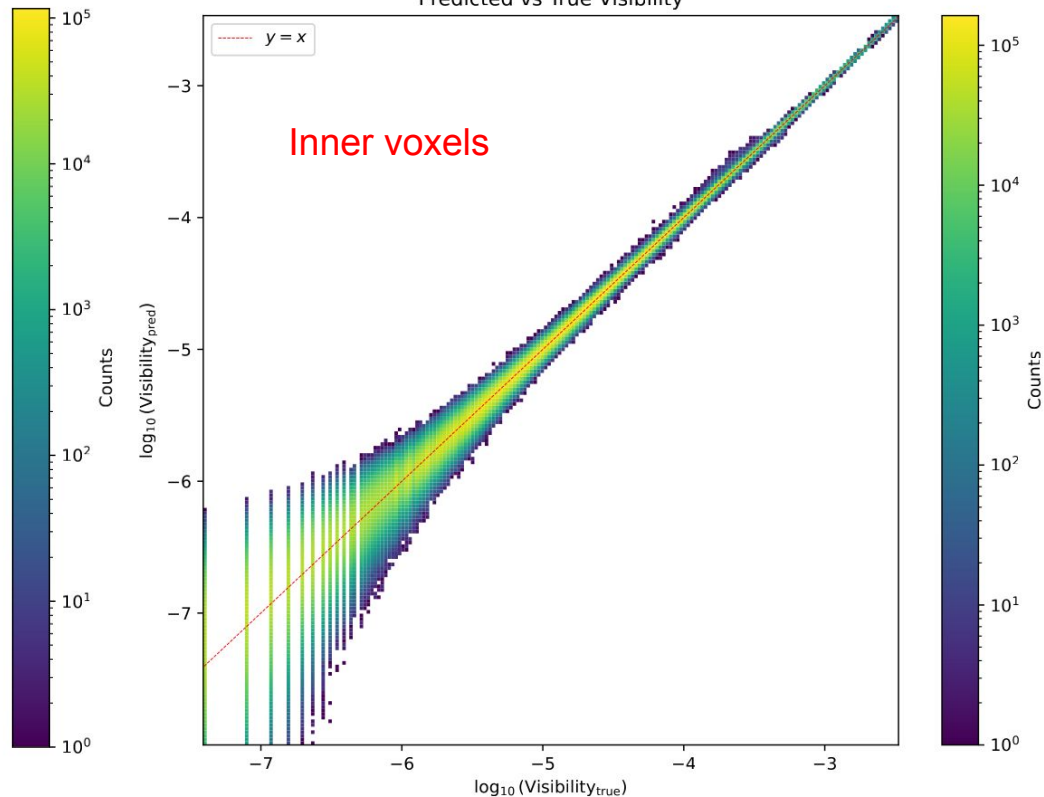
Chi² loss

Predicted vs True Visibility



MSE loss

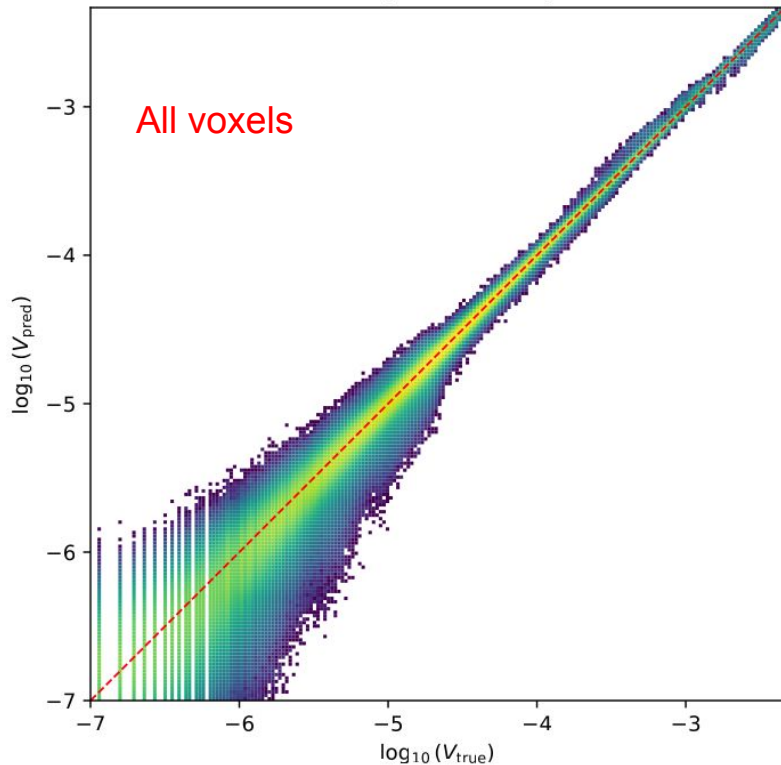
Predicted vs True Visibility



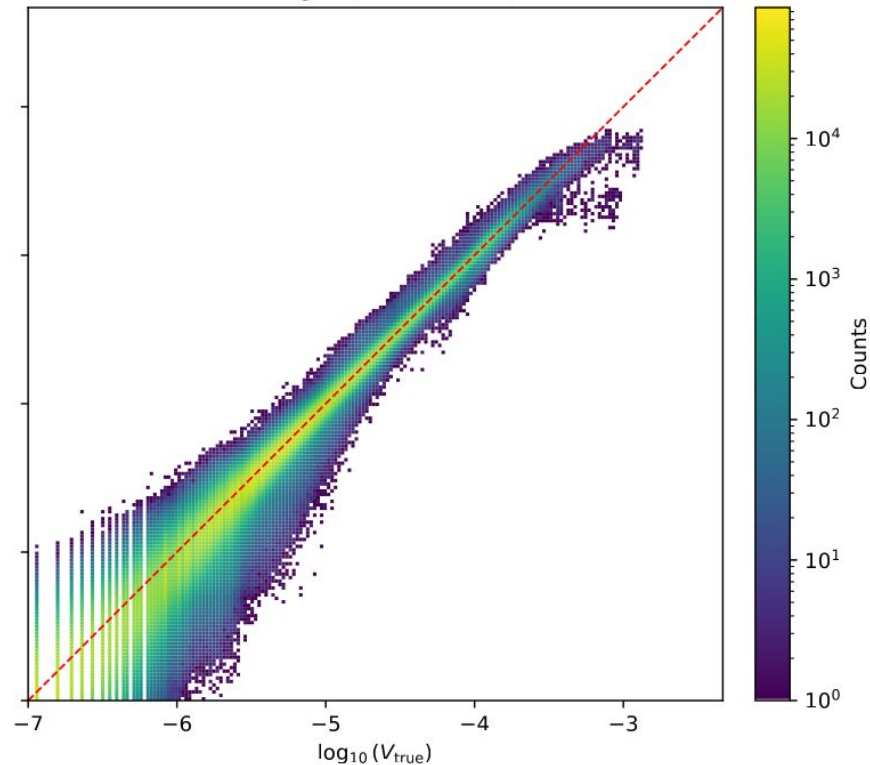
Chi² loss: ACL & LCM

Predicted vs True Visibility

LCM (60 channels)



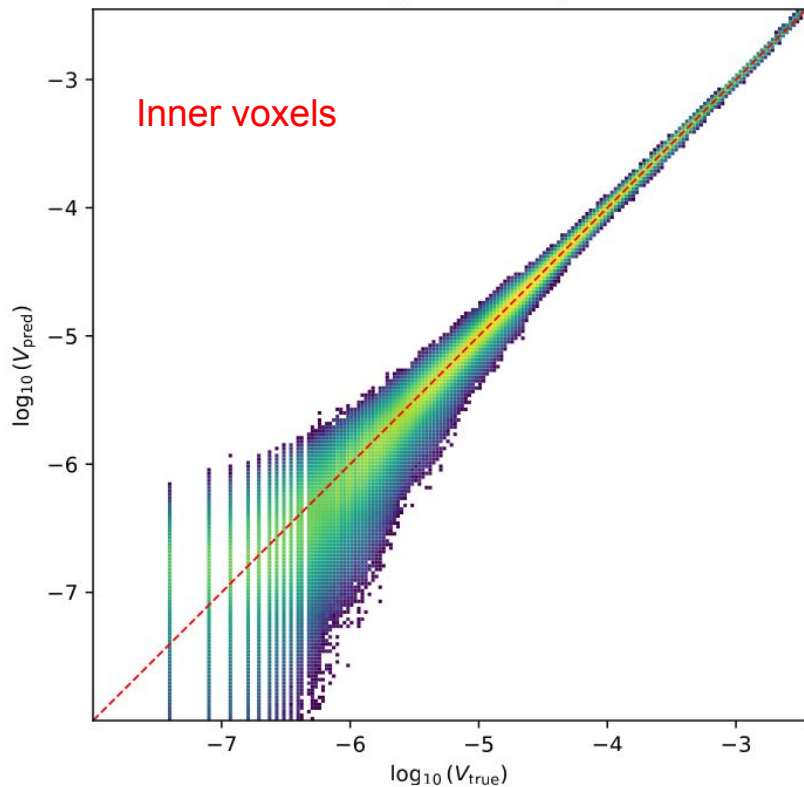
ArCLight (60 channels)



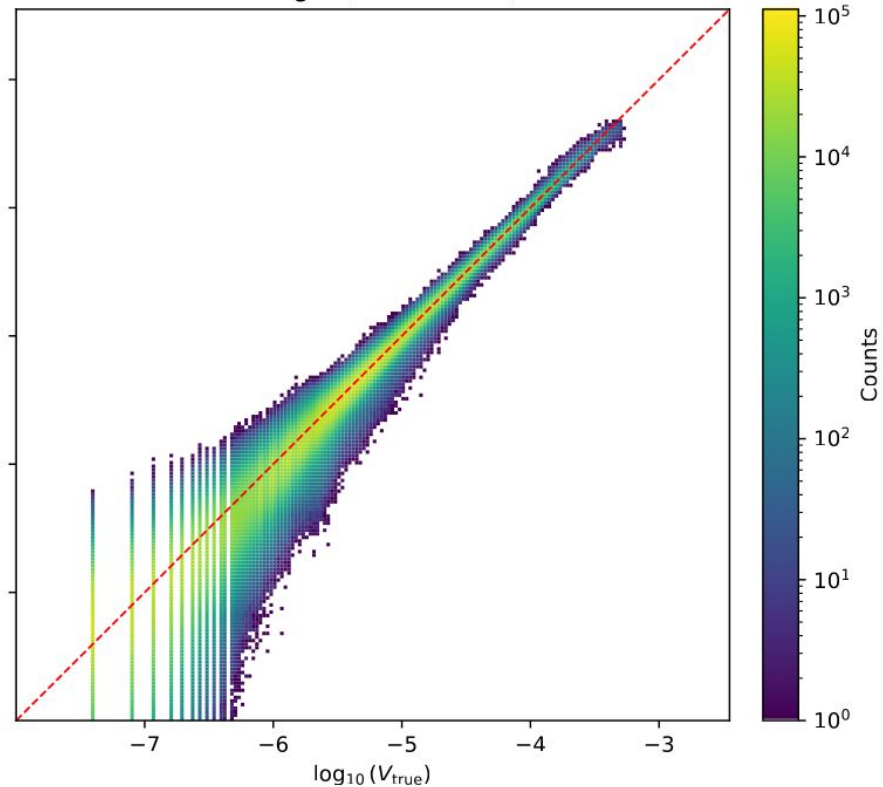
Chi² loss: ACL & LCM

Predicted vs True Visibility

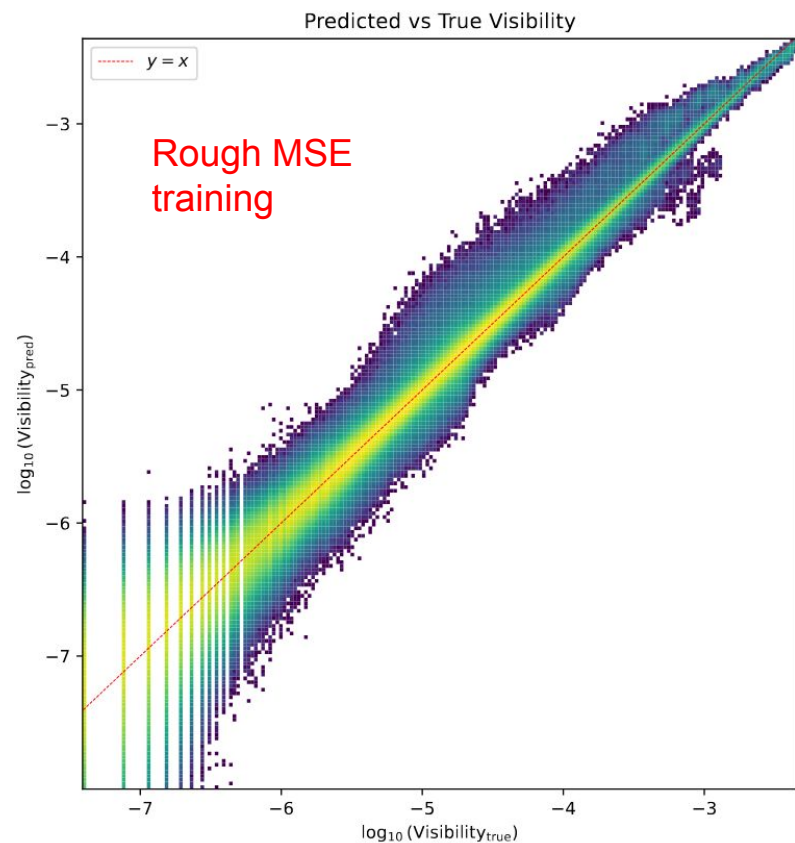
LCM (60 channels)



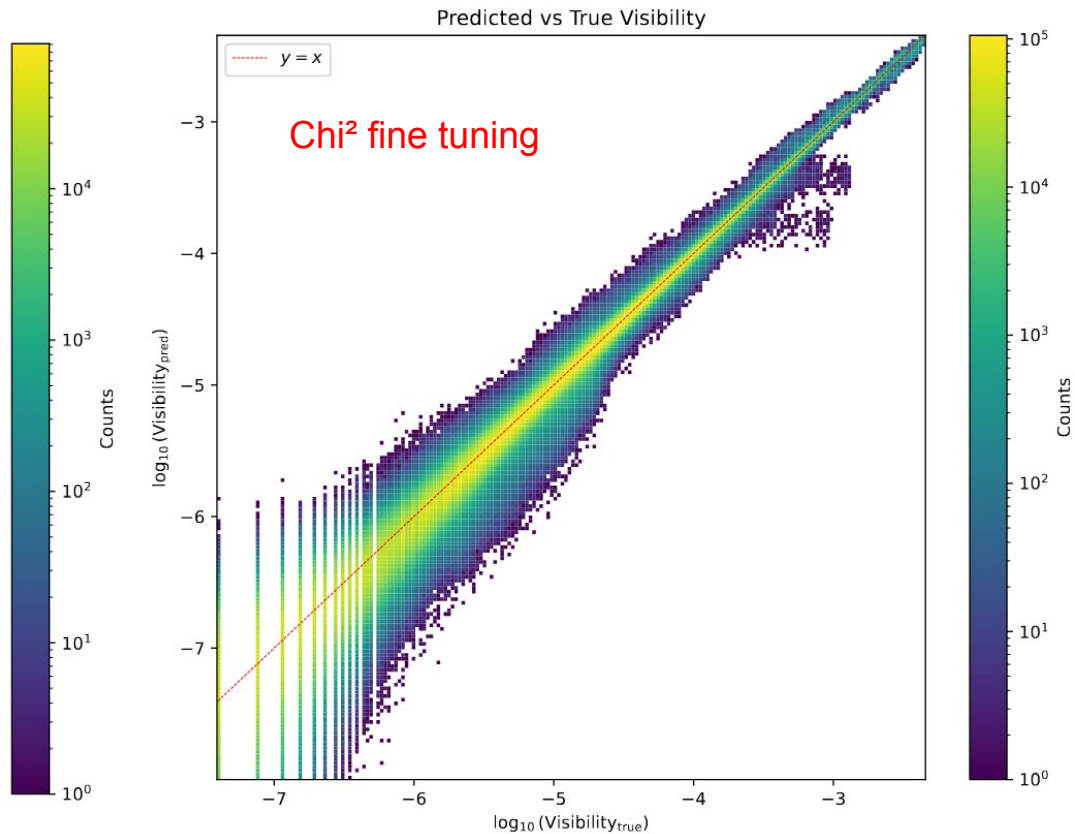
ArCLight (60 channels)



Initialize with MSE, then finetune with χ^2 loss



12.08.2026

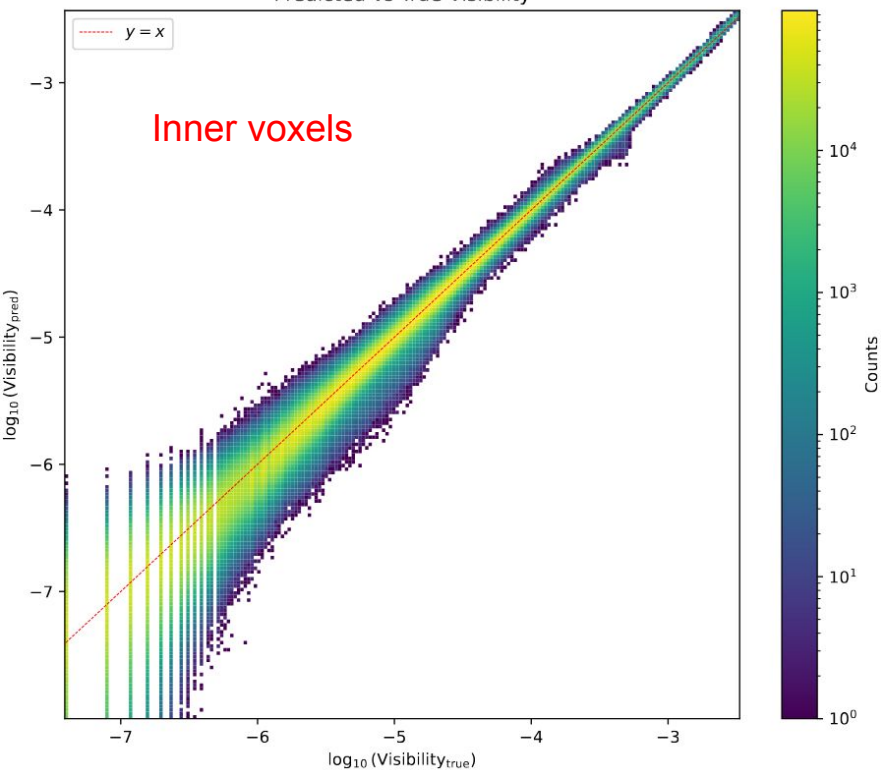


2x2/ML Meeting

20

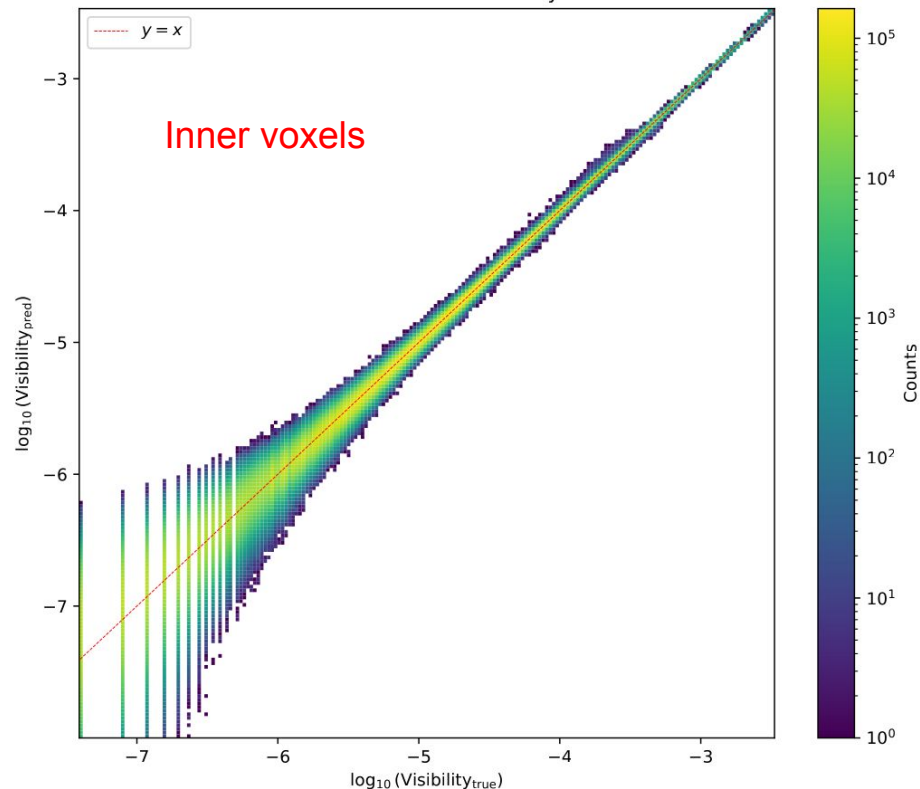
Chi² fine tuning (2k + 2k epochs)

Predicted vs True Visibility



MSE loss (10k epochs)

Predicted vs True Visibility



Next

- Data preparation of muon selection in FSD
- Leave out corners/LT-surface voxels for model training

Implementation of Poisson-loss (?)

Poisson loss:

$$\mathcal{L}_{\text{Poisson}} = \lambda - k \log(\lambda)$$

where

$$k_{\text{true}} = v_{\text{true}} N_{\gamma}$$

$$\lambda_{\text{pred}} = v_{\text{pred}} N_{\gamma}$$

$$\mathcal{L} = \frac{1}{N_{\text{vox}} N_{\text{ch}}} \sum_i \sum_c [\lambda_{i,c} - k_{i,c} \log(\lambda_{i,c})]$$