Parzen-PNN Gaussian Mixture Estimator: Results Report

Fabrizio Benvenuti

February 7, 2026

1 Results

1.1 Parzen Window

In this section are reported the estimation ValNLL and MSE obtained by varying the input parameters of the Parzen Window estimator.
(i.e., window size h_1 and number of sampled points per gaussian in the mixture).

1.1.1 Parzen Window Errors

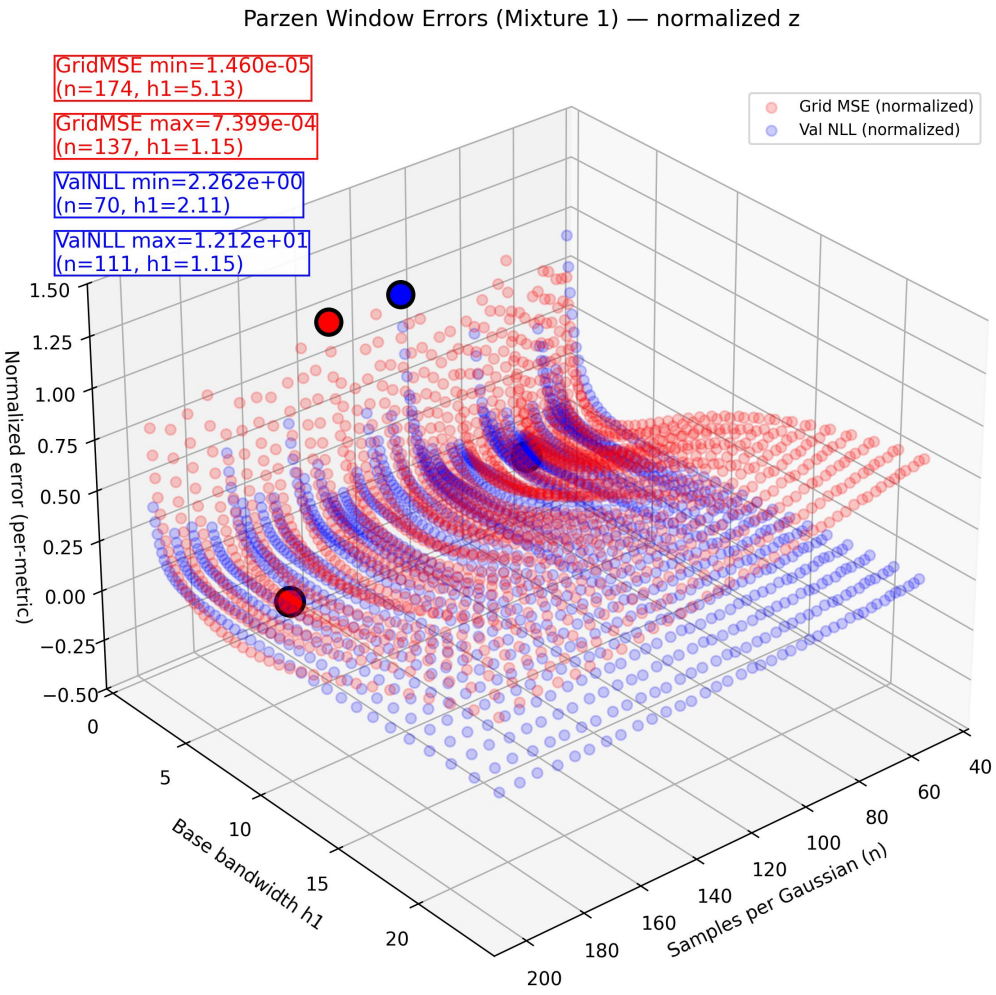

Figure 1: In red: MSE between the mixture 1 pdf and its PW estimate
In blue: ValNLL between PW estimate and the sampled points;
while varying the window size h_1 and the number of sampled points for each gaussian.

In this graph it's marked how the MSE does not vary linearly with window size;
approaching zero at the optimal value of h_1 and increasing exponentially when undersmoothing occurs;
transitioning to the oversmoothed region, the MSE still increases but less steeply.
Samples per gaussian seem to have an exponential impact on reducing MSE,
even if its effect is less visible than the base bandwidth one.
NLL does show a lower steepness than the MSE, in the oversmoothing region.

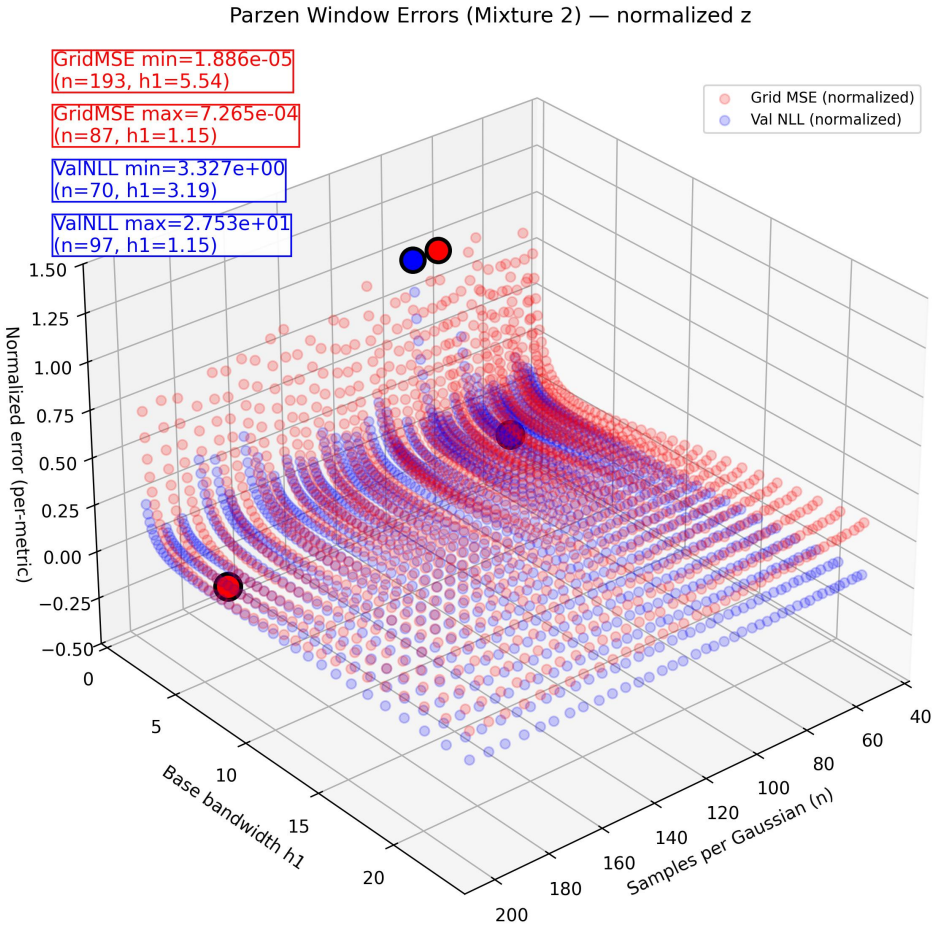

Figure 2: In red: MSE between the mixture 2 pdf and its PW estimate
In blue: ValNLL between PW estimate and the sampled points;
while varying the window size h_1 and the number of sampled points for each gaussian.

In this graph it's marked how the MSE steepnes in the oversmoothed region is less pronounced compared to mixture 1; this is probably due to the fact that oversmothing the probability mass causes less error when the modes are closer together; whilst, in the mixture 1, overmoothing causes the probability mass to fall on the tails; which generates more error.

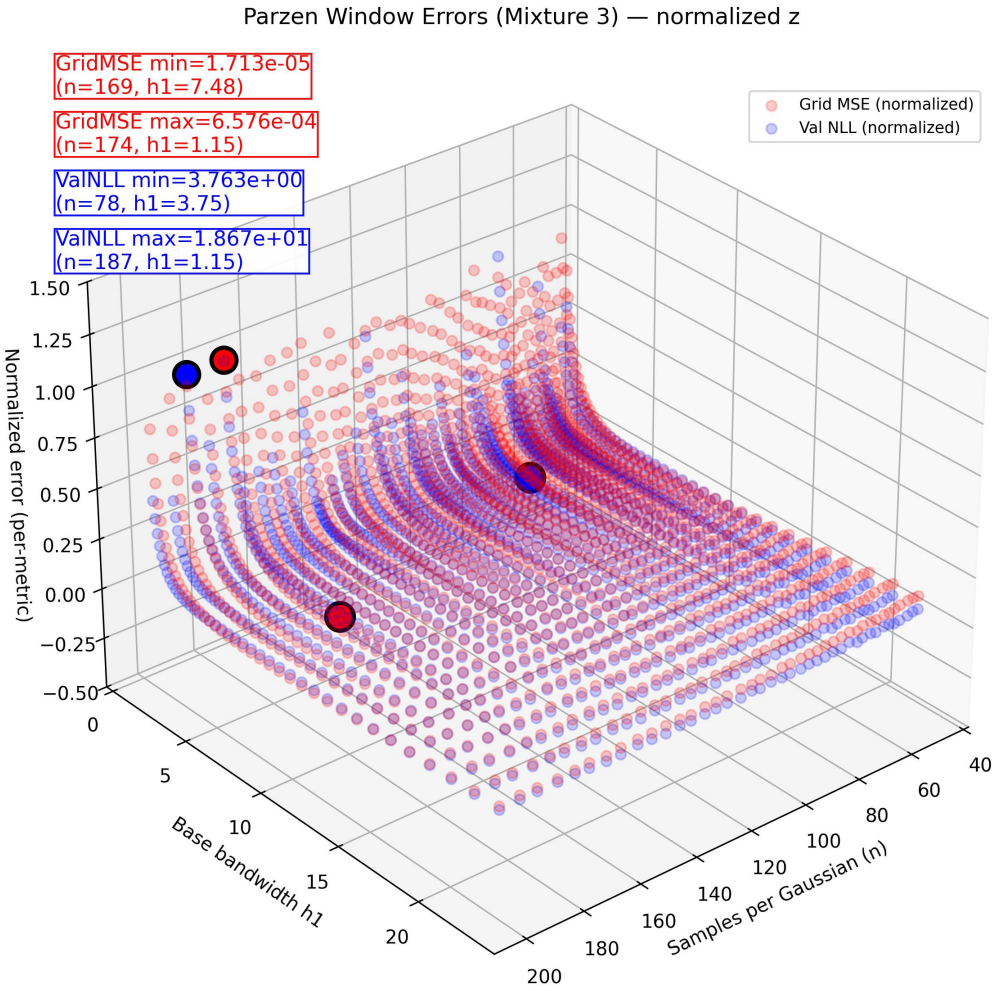

Figure 3: In red: MSE between the mixture 3 pdf and its PW estimate
In blue: ValNLL between PW estimate and the sampled points;
while varying the window size h_1 and the number of sampled points for each gaussian.

In this graph it's noticeable how increasing sampled points seems to have
less impact on reducing MSE compared to mixture 1 and 2.
It's also easibly determinable how the NLL curve does not seem to change shape nor
optimal h_1 between mixtures as much as the MSE curve does.
This causes the selected mixtures to be undersmoothed, the more gaussians there are in the mixture.

$$h^{\text{opt}}_{1,\text{MSE}}(\text{mix}_1, \text{mix}_2, \text{mix}_3) = (5.13,\ 5.54,\ 7, 48), \qquad h^{\text{opt}}_{1,\text{NLL}}(\text{mix}_1, \text{mix}_2, \text{mix}_3) = (2.11,\ 3.19,\ 3.75).$$

1.1.2 Parzen Window Overlays

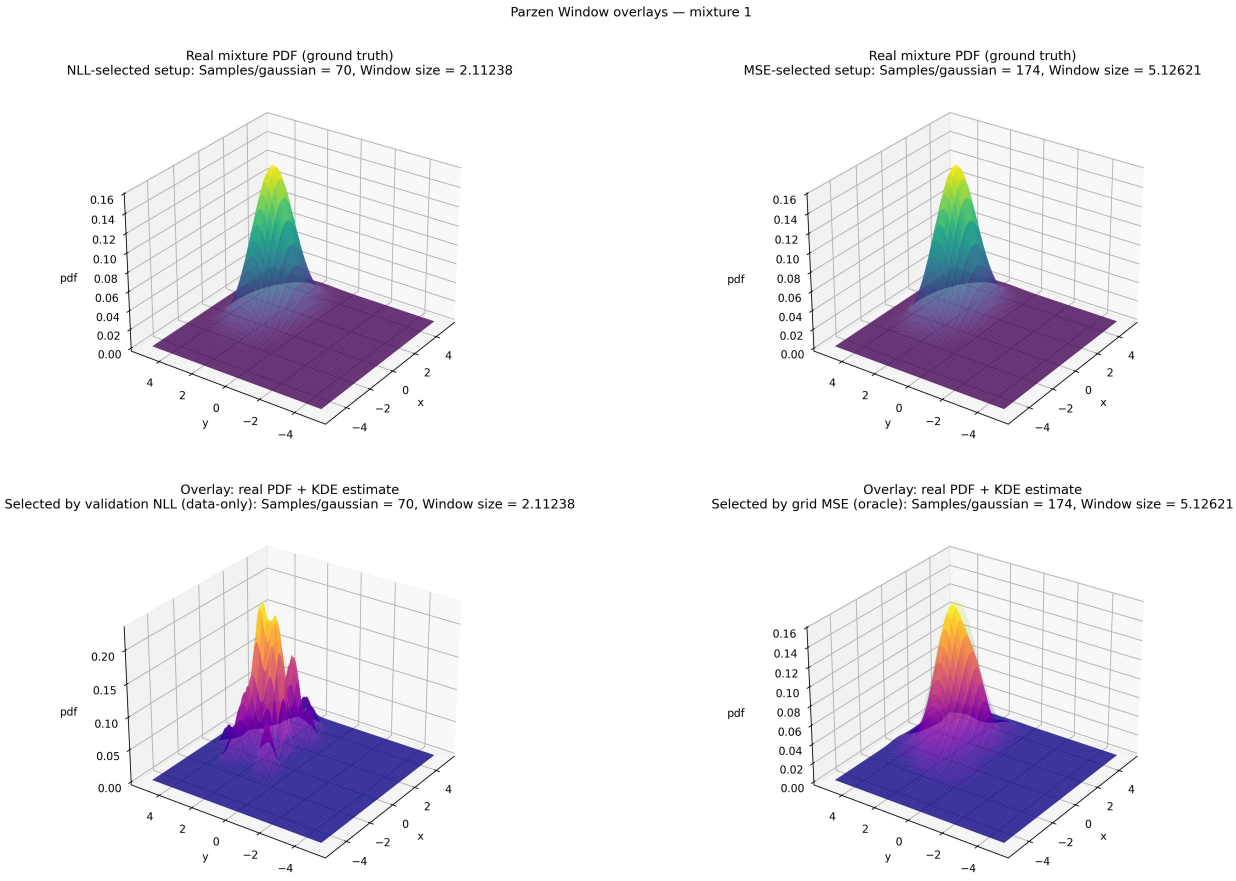

Figure 4: Top row: pdf mixture 1 pdf;
Bottom: PW estimates selected by minimizing NLL and MSE respectively.

In this graph it's noticeable how the min MSE was selected with
higher points per gaussian than the NLL ones.
Therefore the selected parameters by NLL are undersmoothed compared to the MSE ones.

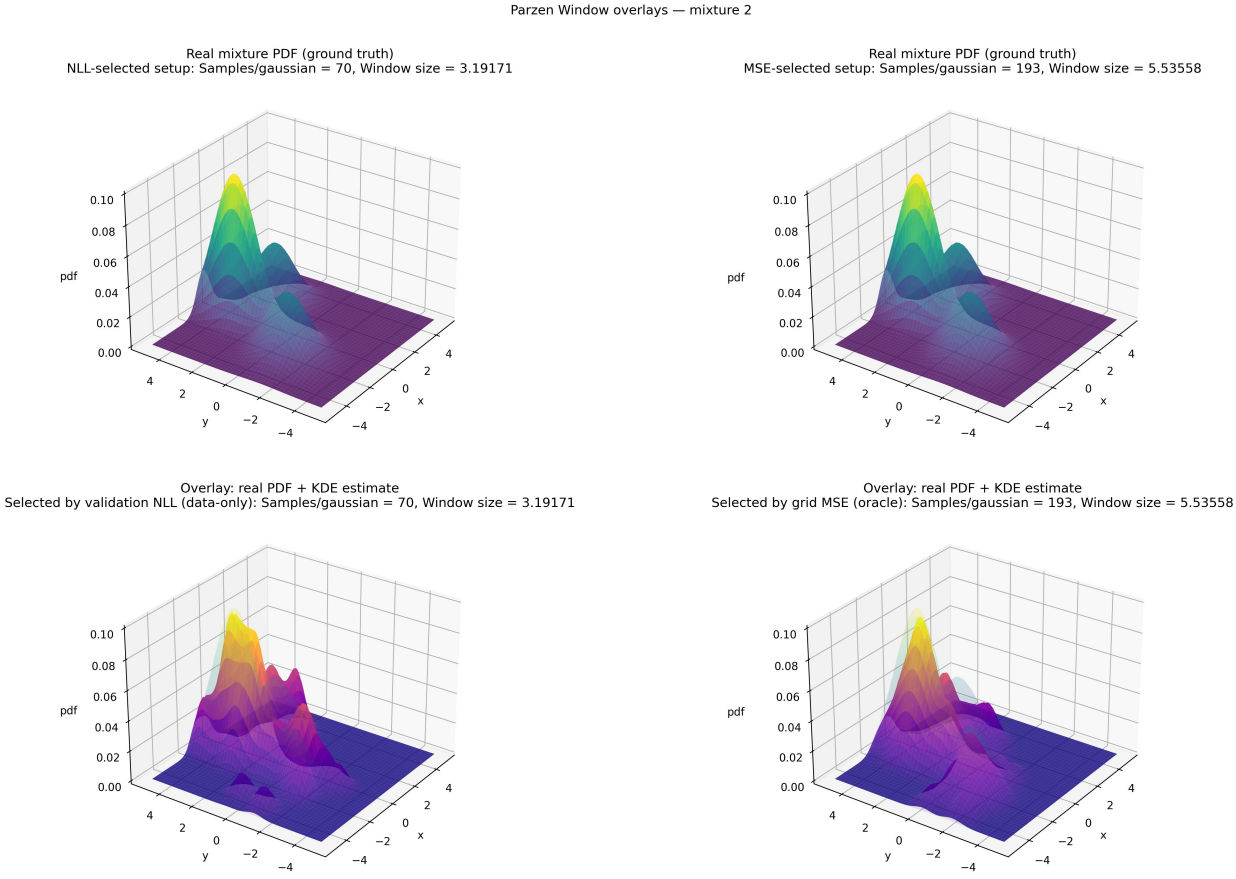

Figure 5: Top row: pdf mixture 2 pdf;
Bottom: PW estimates selected by minimizing NLL and MSE respectively.

It's also noticeable how the MSE selected parameters still cannot provide the best accuracy in estimating the pdf's gaussians with very different variances, this could be due to the fact that $h_n = \frac{h_1}{\sqrt{n-1}}$ does not converge to 0 so fast ???.

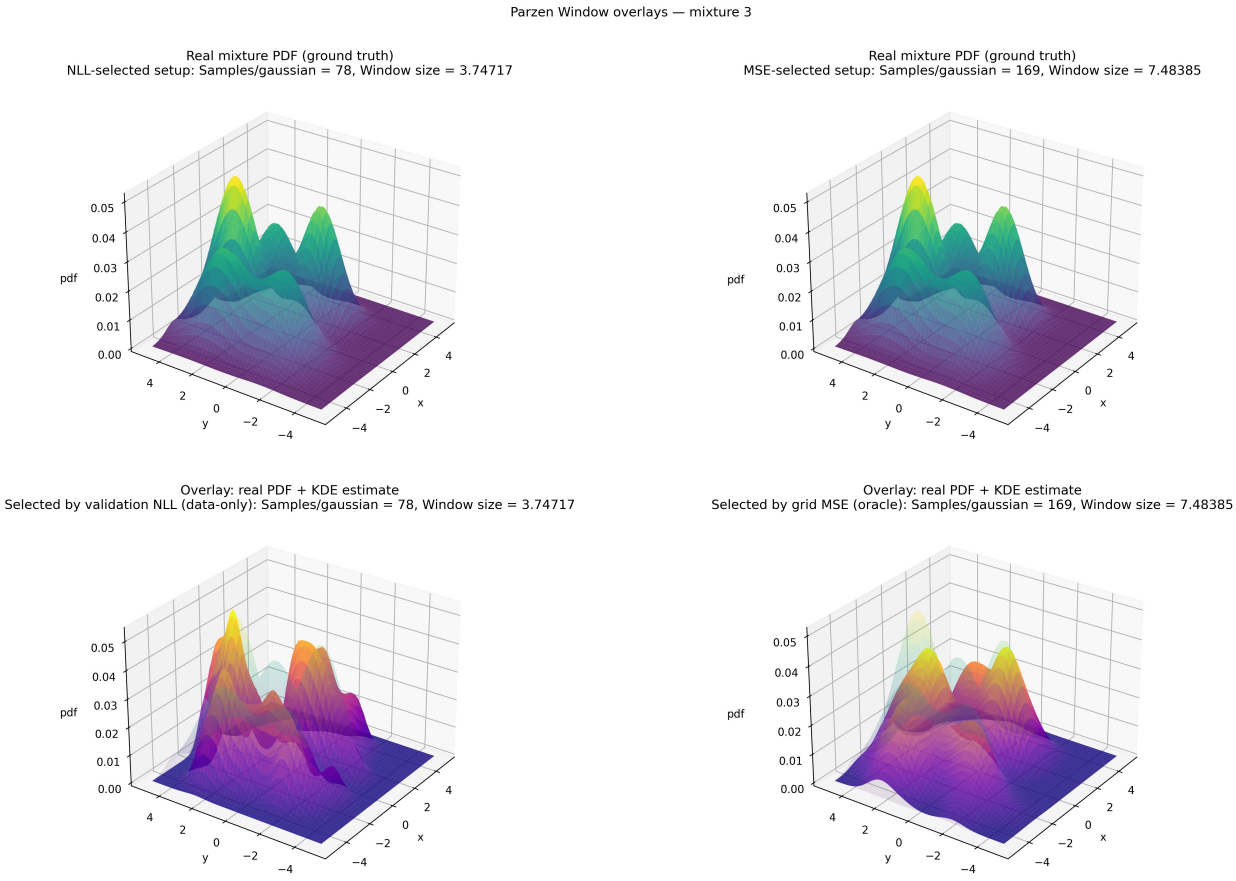

Figure 6: Top row: pdf mixture 3 pdf;
Bottom: PW estimates selected by minimizing NLL and MSE respectively.

In this graph it's instead noticeable how NLL selected parameters
may sometime lead the predicted pdf with undersmoothed regions;
that approximates single peaks as two distinct modes.
This could even be whilst having a low enough NLL value; because data points drawn
from those peaks will still have high likelihood even if there is a valley in between them.
This effect is less visible in MSE selected parameters since the overall shape of the pdf
is more important than the likelihood of single data points.

1.2 Parzen Neural Network

1.2.1 Parzen Neural Network Errors

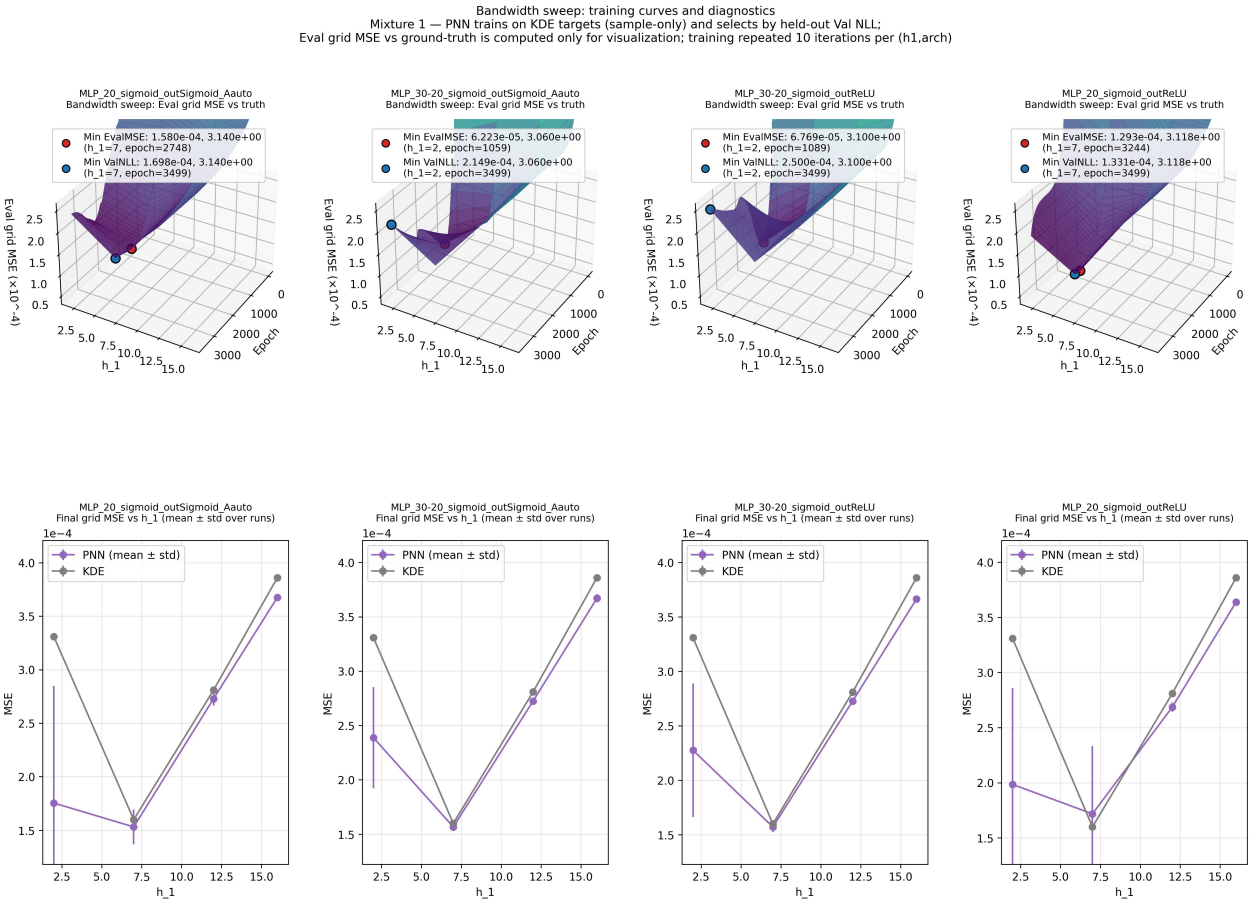

Figure 7: Top: Eval-MSE surface between mixture 1 pdf and PNN estimate over epoch × h_1
Bottom: final grid MSE (last epoch) vs h_1 shown as mean ± std
across 10 runs (PNN) and KDE for reference.

In this graphs it's noticeable how the PNNs tends to have a lower optimal bandwidth with respect to the PW.
which could also be seen in the error subgraphs with MSE vs h_1,
where it is obvious that the PNN seems to work better than the PW with undersmoothed KDEs.
It is also marked how the the MSE's std, in the undersmoothed region,
for deep architectures is quite smaller than in the single layered PNNs.

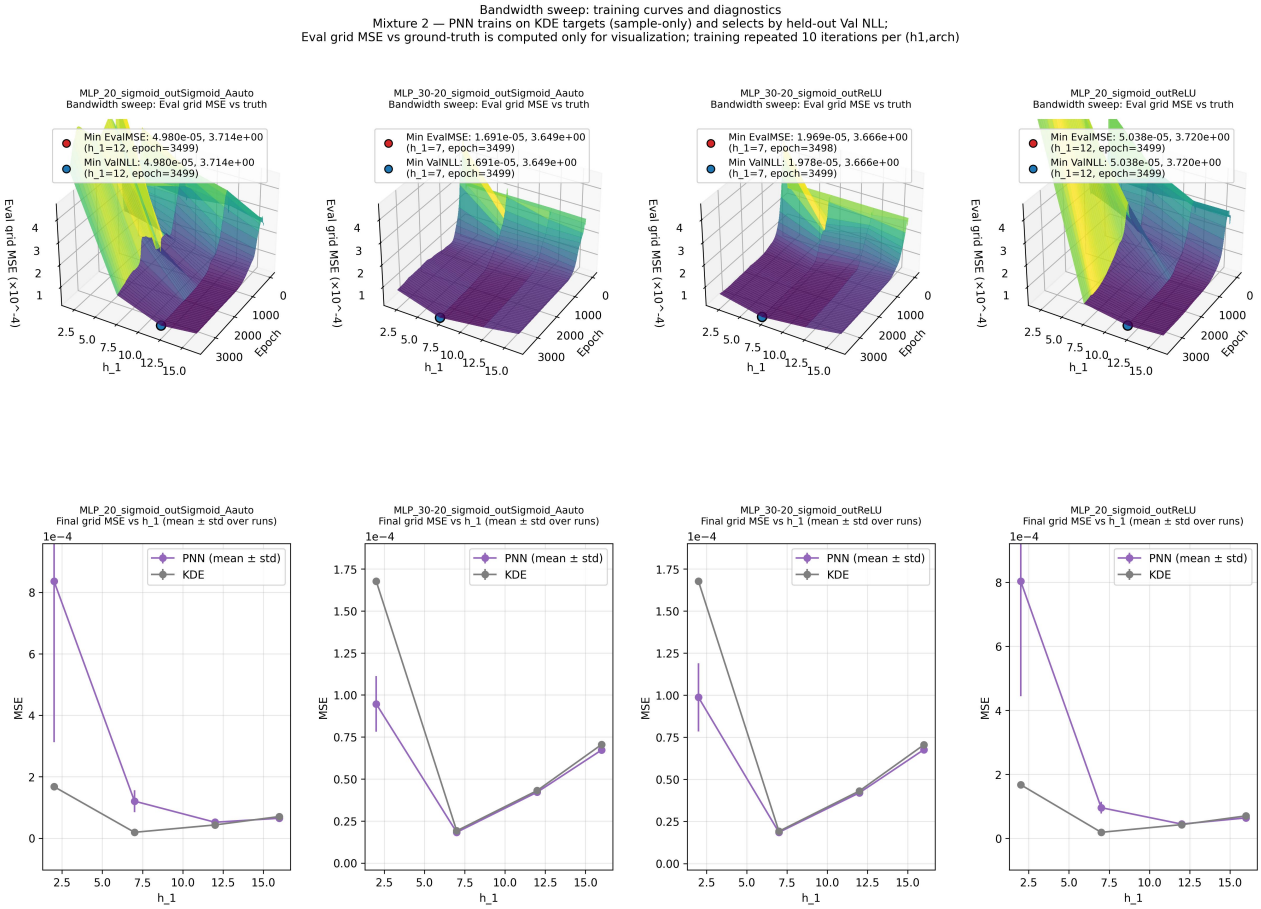

Figure 8: Top: Eval-MSE surface between mixture 2 pdf and PNN estimate over epoch × h_1
Bottom: final grid MSE (last epoch) vs h_1 shown as mean ± std
across 10 runs (PNN) and KDE for reference.

In this graph it's noticable how increasing the number of gaussians in the mixture causes:

- Single hidden layer PNNs to perform worse than KDEs and double hidden layered PNNs, whilst beeing extremely sensitive to h_1 variations, especially in the undersmoothed region.
- The MSE mesh seems to be flattened, especially in the oversmoothed region, with respect to the mixture1.
- In this graph it's marked how PNNs resiliency to undersmoothed KDEs increases, in the deep architectures, with the increase of gaussians inside the mixture.

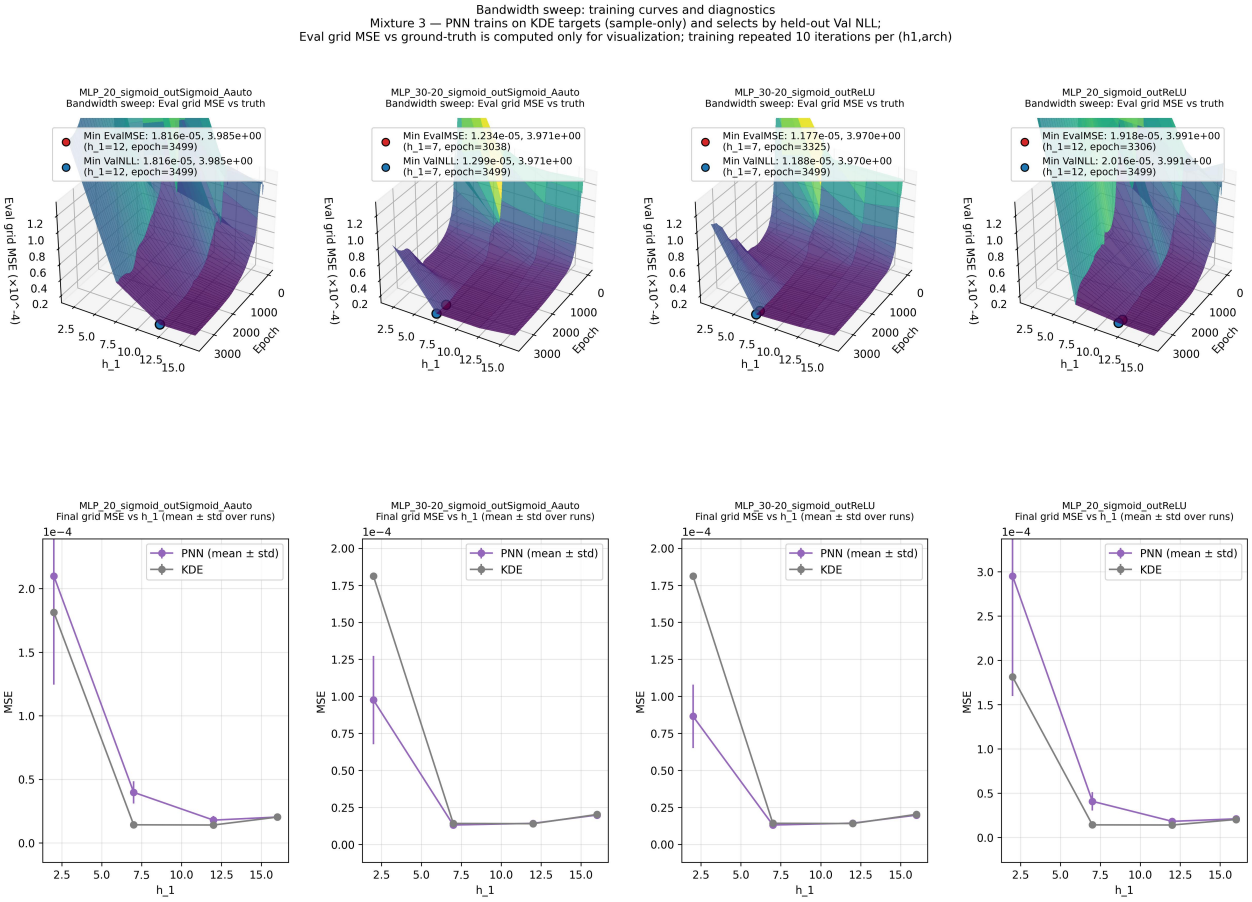

Figure 9: Top row: Eval-MSE surface between mixture 3 pdf and PNN estimate over epoch × h_1
Bottom row: final grid MSE (last epoch) vs h_1 shown as mean ± std
across 10 runs (PNN) and KDE for reference.

It's marked in this graphs how increasing the number of points inside the PDF (by increasing the number of gaussians inside the mixture); causes the Val-NLL (based onto held-out data points) to be a good metric to determine the best combination of architecture, and KDE bandwidth. This could be seen because the mixture 3 is the one where Val-NLL points have the lowest EvalMSE, across all mixtures.

1.2.2 Parzen Neural Network Overlays

In this subsection are shown the overlays at minimum validation NLL (ValNLL), for each mixture and each architecture.
Each column corresponds to one architecture and is displayed at its own best bandwidth h_1 (the one minimizing ValNLL for that architecture on the held-out points).

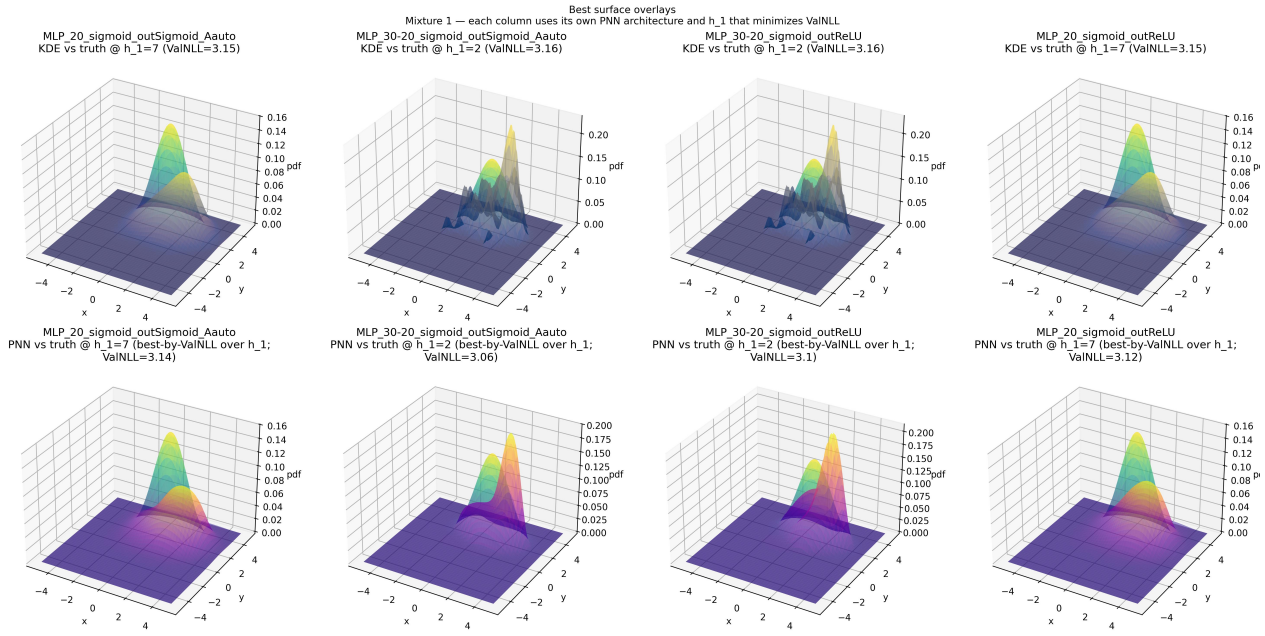

Figure 10: Best overlays for mixture 1, selected by minimizing ValNLL.
Top row: Parzen Window (KDE) estimate vs ground-truth mixture PDF at the same selected h_1.
Bottom row: Parzen Neural Network (PNN) estimate vs ground-truth mixture PDF.

In this graph it is appreciable how PNN overlays from deep architectures have an easyer time at approximating high variance sections.
and how using held-out points for validation causes selection of smoothed out overlays even with target KDEs are undersmoothed.
??why the single hidden layered architectures chose oversmoothed KDEs
and still got comparable valNLL even if it is widely different than the target pdf??

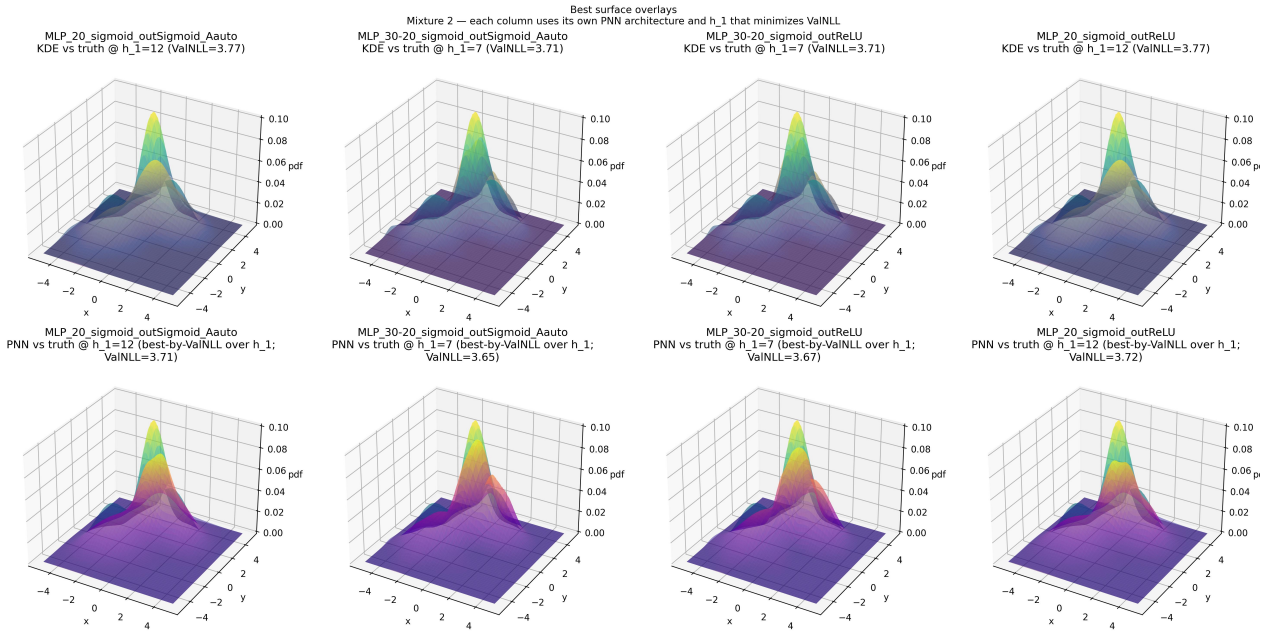

Figure 11: Best overlays for mixture 2, selected by minimizing ValNLL.
Top row: Parzen Window (KDE) estimate vs ground-truth mixture PDF at the same selected h_1.
Bottom row: Parzen Neural Network (PNN) estimate vs ground-truth mixture PDF.

In this graph it's instead appreciable how, even with oversmoothed KDEs
using the ValNLL as a loss function, can restore the high variance regions on the PNN estimate??.

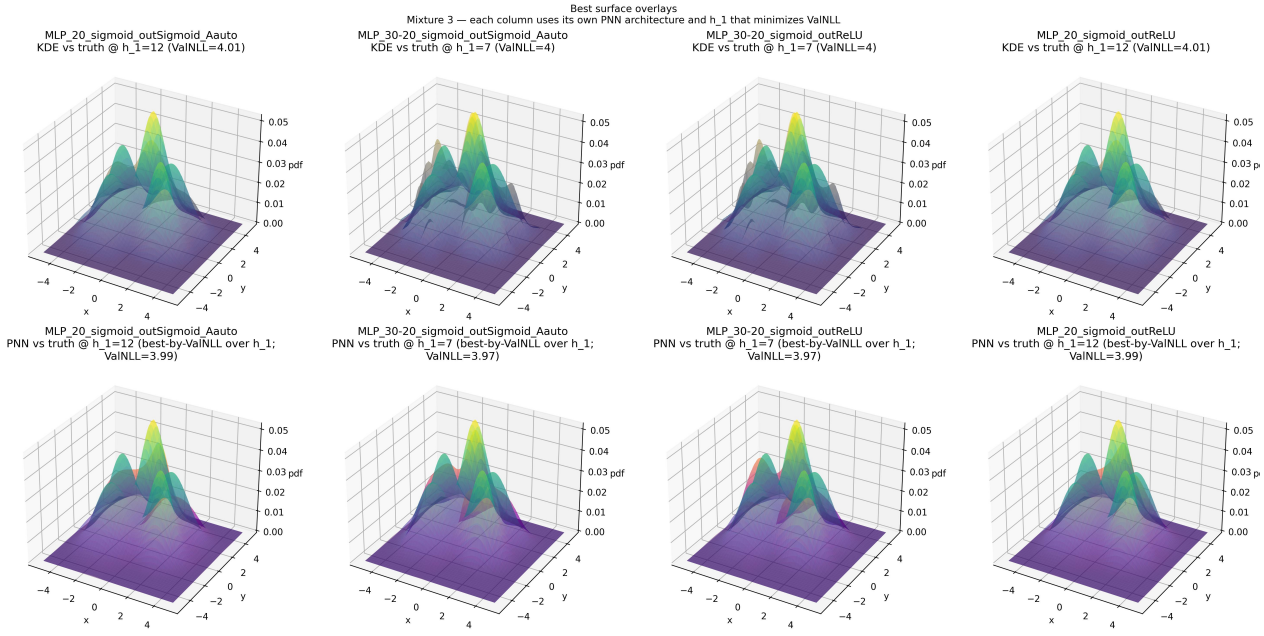

Figure 12: Best overlays for mixture 3, selected by minimizing ValNLL.
Top row: Parzen Window (KDE) estimate vs ground-truth mixture PDF at the same selected h_1.
Bottom row: Parzen Neural Network (PNN) estimate vs ground-truth mixture PDF.

It's marked how in this graphs the PNNs with higher layer count still have a hard time at approximating high variance peaks close to each others in the pdf, this could be due to the fact that given that most points are inside the R_n region inside the peak, the MLE

that estimates points between the peaks, as a 'ridge' between the peaks, still gets overall low NLL because those points are few, compared to the ones inside the peaks

1.2.3 Parzen Neural Network Boundary strategy evaluation

In this subsection are underlined the comparisons on how using the support X of the sampled points to give penalty for estimations with 'heavy-tails' changes the shape and ValNLL of the estimates.

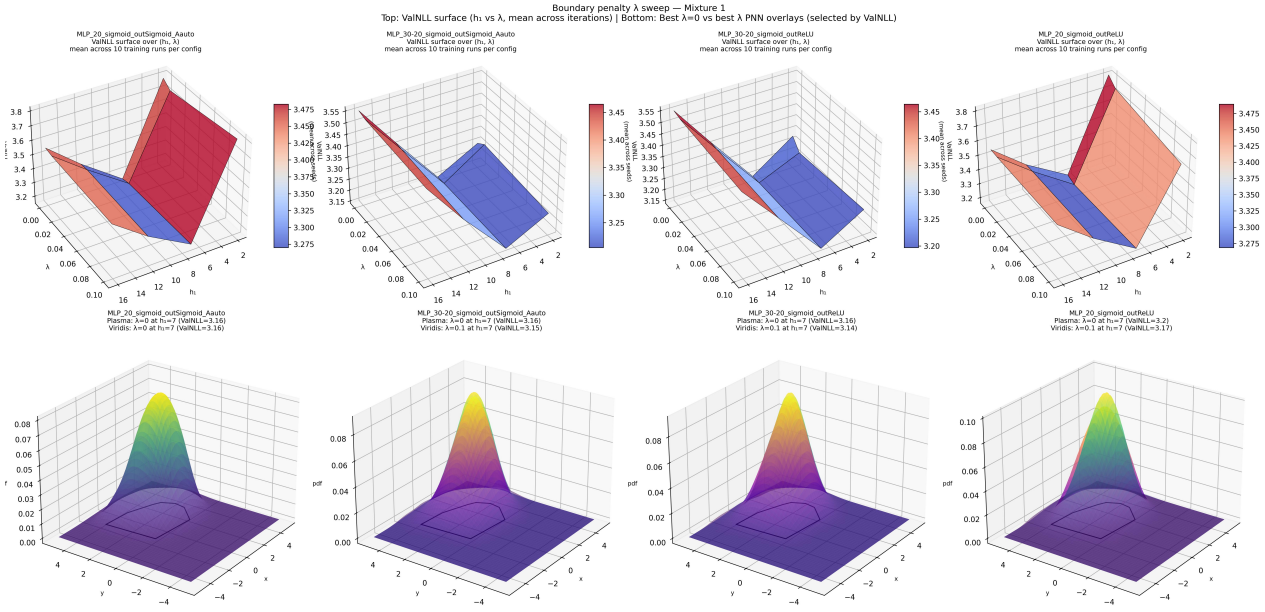

Figure 13: Top Row: ValNLL for mixture 1 vs $h_1 \times \lambda$ for every PNN architecture
Bottom Row: PNN overlay that minimizes ValNLL@$\lambda = 0$
vs the PNN that minimizes ValNLL $\forall\lambda$

In this graph is instead clear how lambda helps to bring down ValNLL especially for heavily oversmoothed gaussian mixtures. This overall helps the gaussians to have a lower ValNLL for larger h_1 but does not help much in decreasing min(ValNLL). In this graphs it's also visible how increasing λ even in highly oversmoothed regions
does not increase ValNLL.

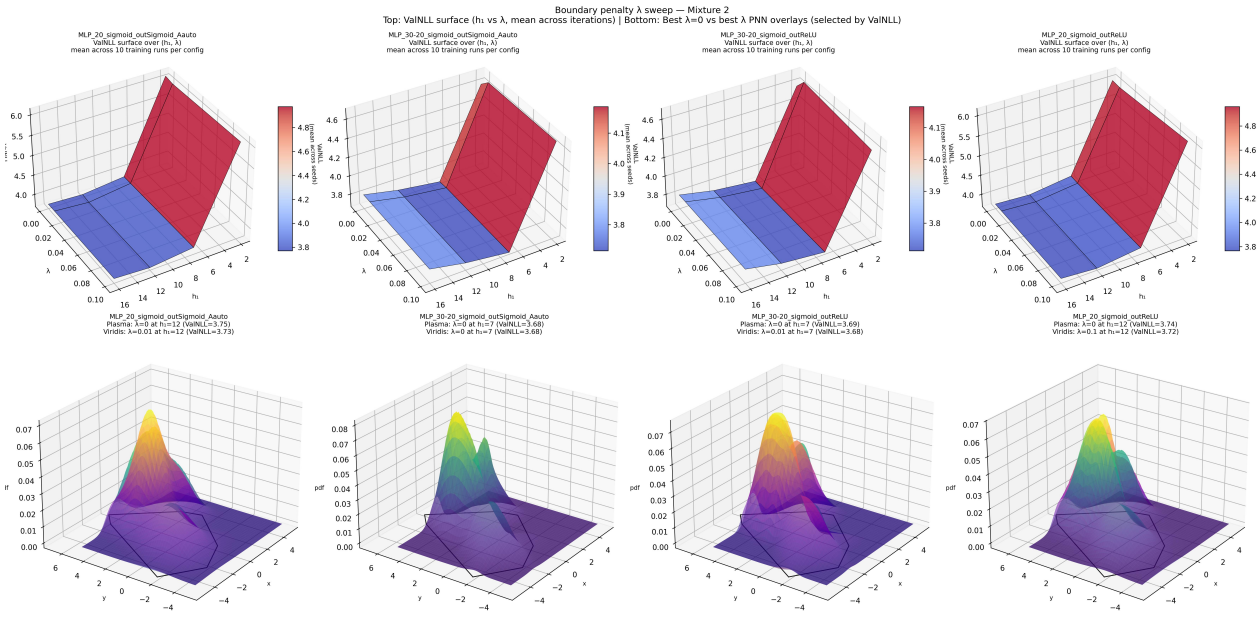

Figure 14: Top Row: ValNLL for mixture 2 vs $h_1 \times \lambda$ for every PNN architecture
Bottom Row: PNN overlay that minimizes ValNLL@$\lambda = 0$
vs the PNN that minimizes ValNLL $\forall\lambda$

In this figure it's clear how heavy tails do not seem to show up as much as in the mixture 1 even with $\lambda = 0$ and heavy undersmoothing, this is due to the fact that increasing the number of gaussians prompts the gaussian to consider near 0 the tails with more confidence????

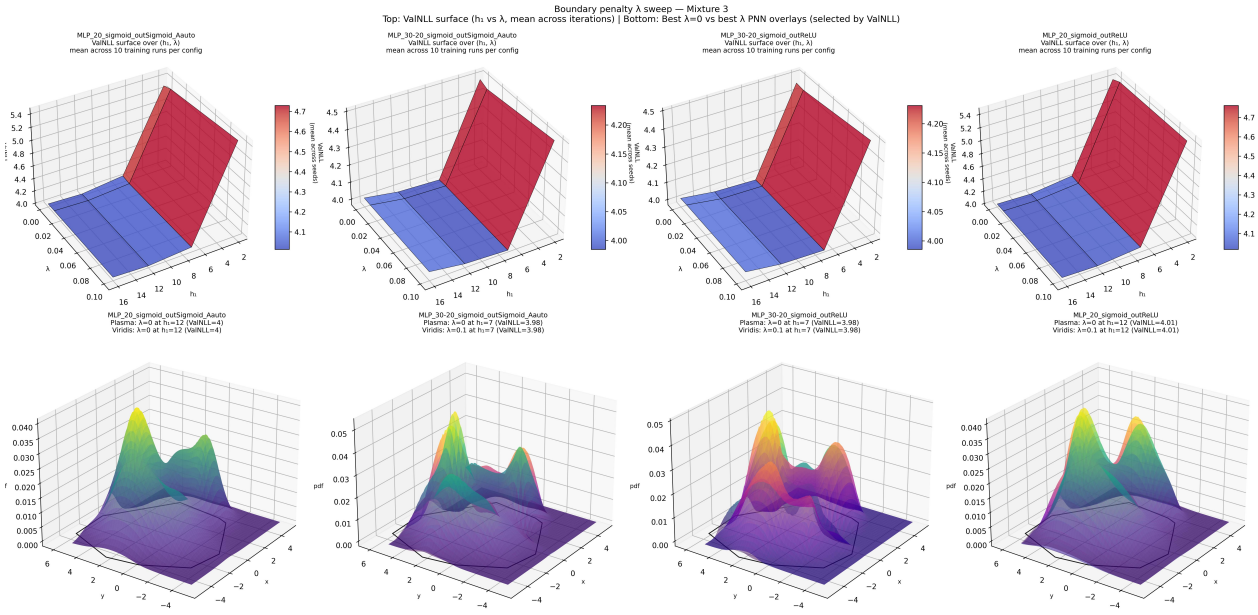

Figure 15: Top Row: ValNLL for mixture 3 vs $h_1 \times \lambda$ for every PNN architecture
Bottom Row: PNN overlay that minimizes ValNLL@$\lambda = 0$
vs the PNN that minimizes ValNLL $\forall\lambda$

It's clear from these graphs that mixtures with high optimal h_1 tend to have havier tails outside the Convex Hull that contains the sampled points, even with high λ probably

lambda should increase more to reduce this fenomenum.