Parzen-PNN Gaussian Mixture Estimator: Results Report

Fabrizio Benvenuti

February 10, 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

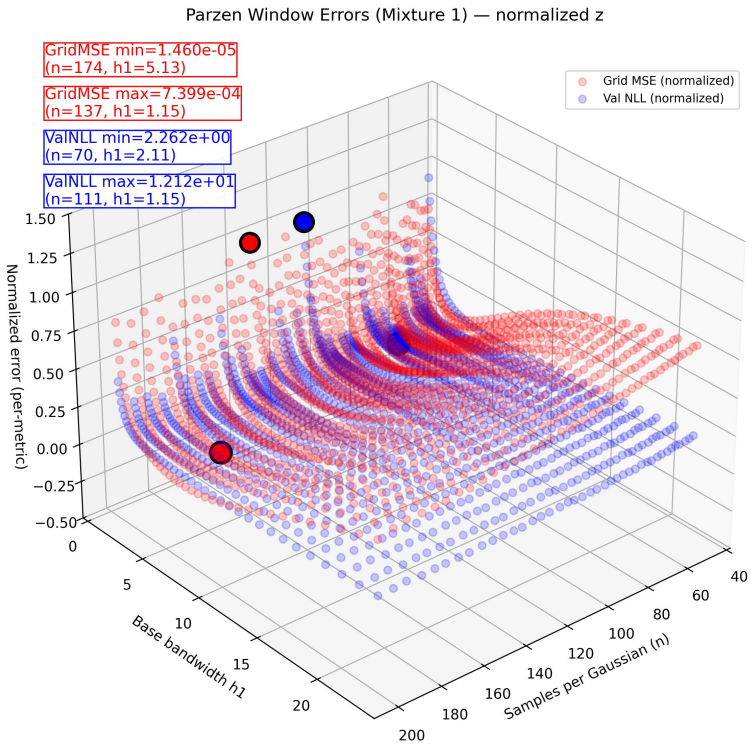

Figure 1: In red: MSE between the mixture 1 pdf and its PW estimate
In blue: ValNLL between PW estimate and the sampled points;
while varying the window size h_1 and the number of sampled points for each gaussian.

It was noted that the MSE forms an asymmetric curve: steeply rising when the window narrows (sharp kernels cause high variance in estimates), but gradually rising when widening (over-smoothed predictions flatten peaks). Increasing sampled points per Gaussian substantially improves accuracy, though gains diminish with additional samples.

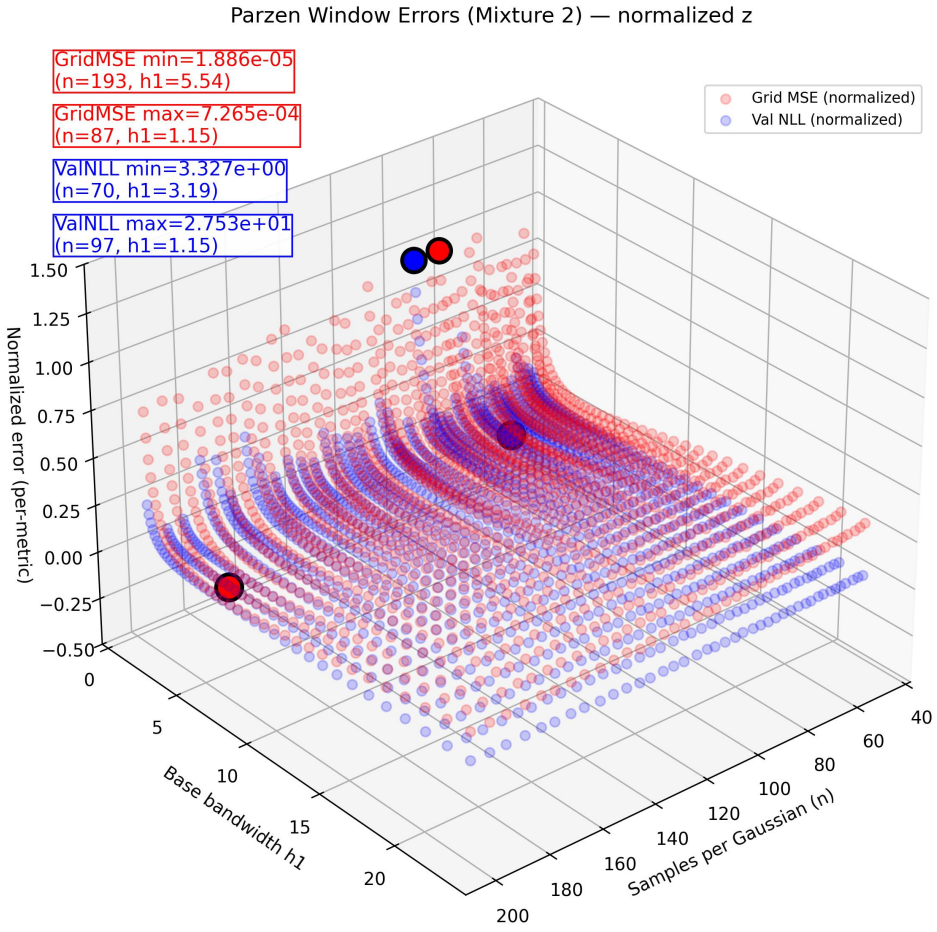

Figure 2: In red: MSE between the mixture 2 pdf and its PW estimate
In blue: ValNLL between PW estimate and the sampled points;
while varying the window size h_1 and the number of sampled points for each gaussian.

It was noted that mixture 2 shows less steep MSE growth in the oversmoothed region compared to mixture 1. This occurs because modes that are close together create a smoother combined density with gentler curvature transitions.
When modes are well-separated, they leave pronounced gaps in between, causing sharp density changes and stronger local curvature.
Such curvature variations directly determine how much error accumulates as the estimate becomes progressively oversmoothed.

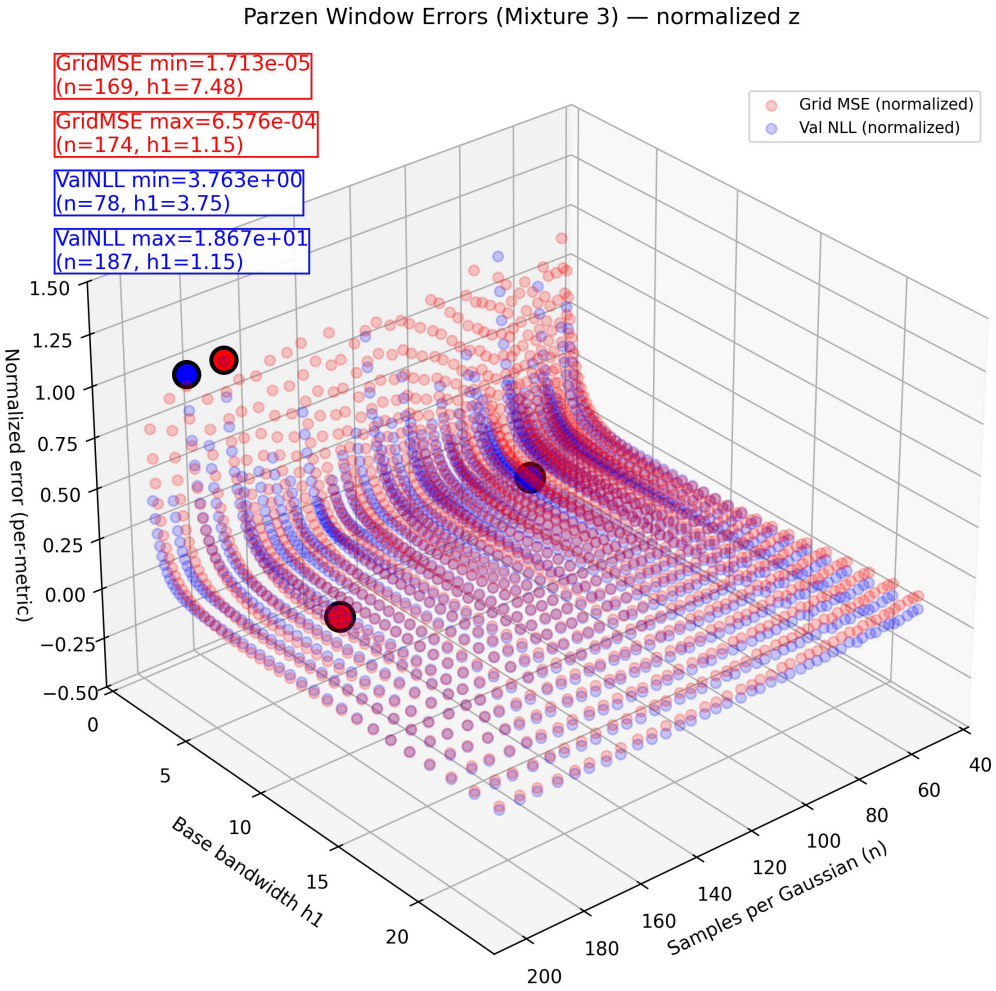

Figure 3: In red: MSE between the mixture 3 pdf and its PW estimate
In blue: ValNLL between PW estimate and the sampled points;
while varying the window size h_1 and the number of sampled points for each gaussian.

It was noticed that Mixture 3's overlapping Gaussians create a smoother density requiring larger MSE-optimal bandwidth, while NLL remains fixated on sharp peaks at sample locations, producing increasingly suboptimal (undersmoothed) density estimates as mixture complexity increases.

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

1.1.2 Parzen Window Overlays

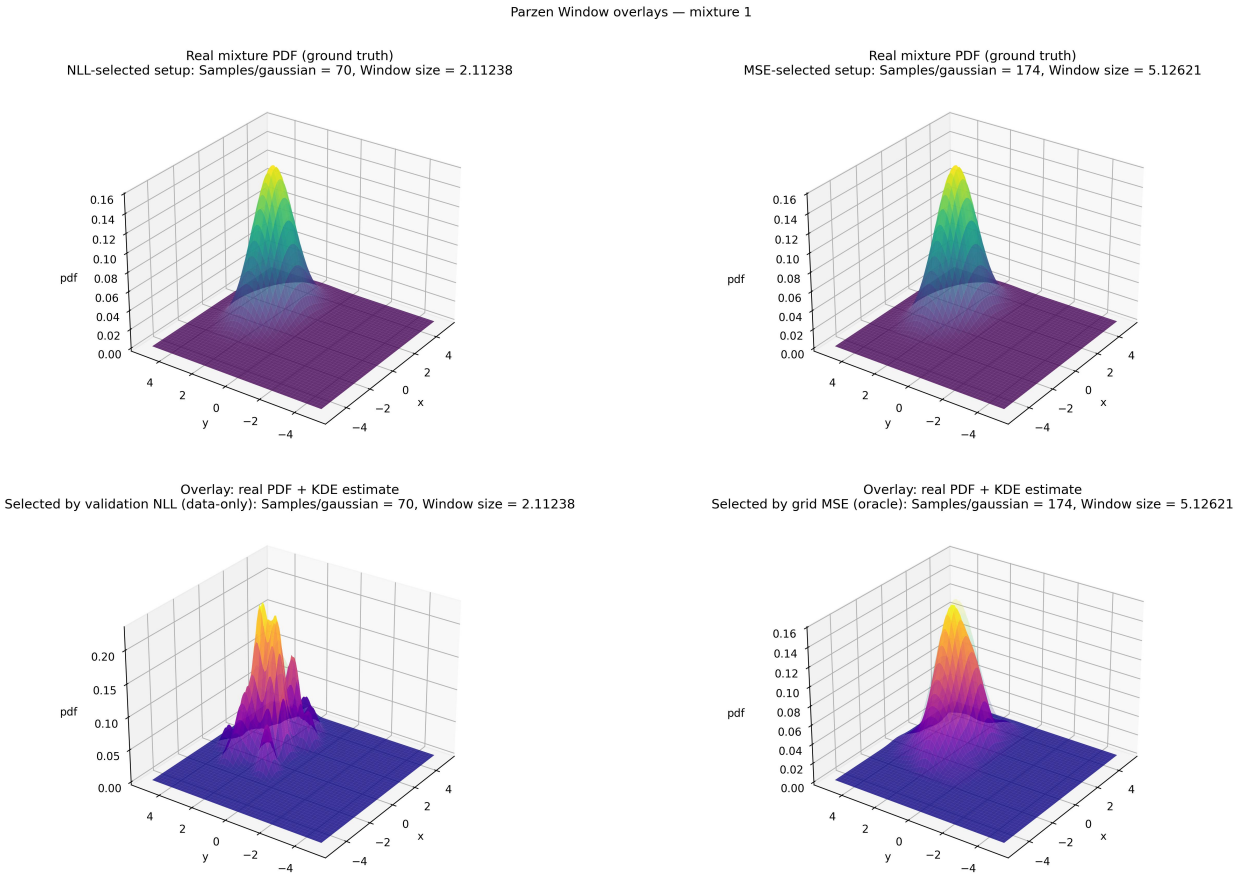

Figure 4: Top row: Mixture 1 pdf;
Bottom: PW estimate overlays selected by minimizing NLL and MSE respectively.

It was marked how minimum MSE selected more points per gaussian compared to NLL, suggesting NLL parameters lean toward undersmoothing. This seems to arise because ValNLL optimization encourages higher peacks where data samples reside, whilst uniform weighting naturally tends toward smoother overall shapes.
The divergence in optimal sample points also seems to amplify these bandwidth disparities.

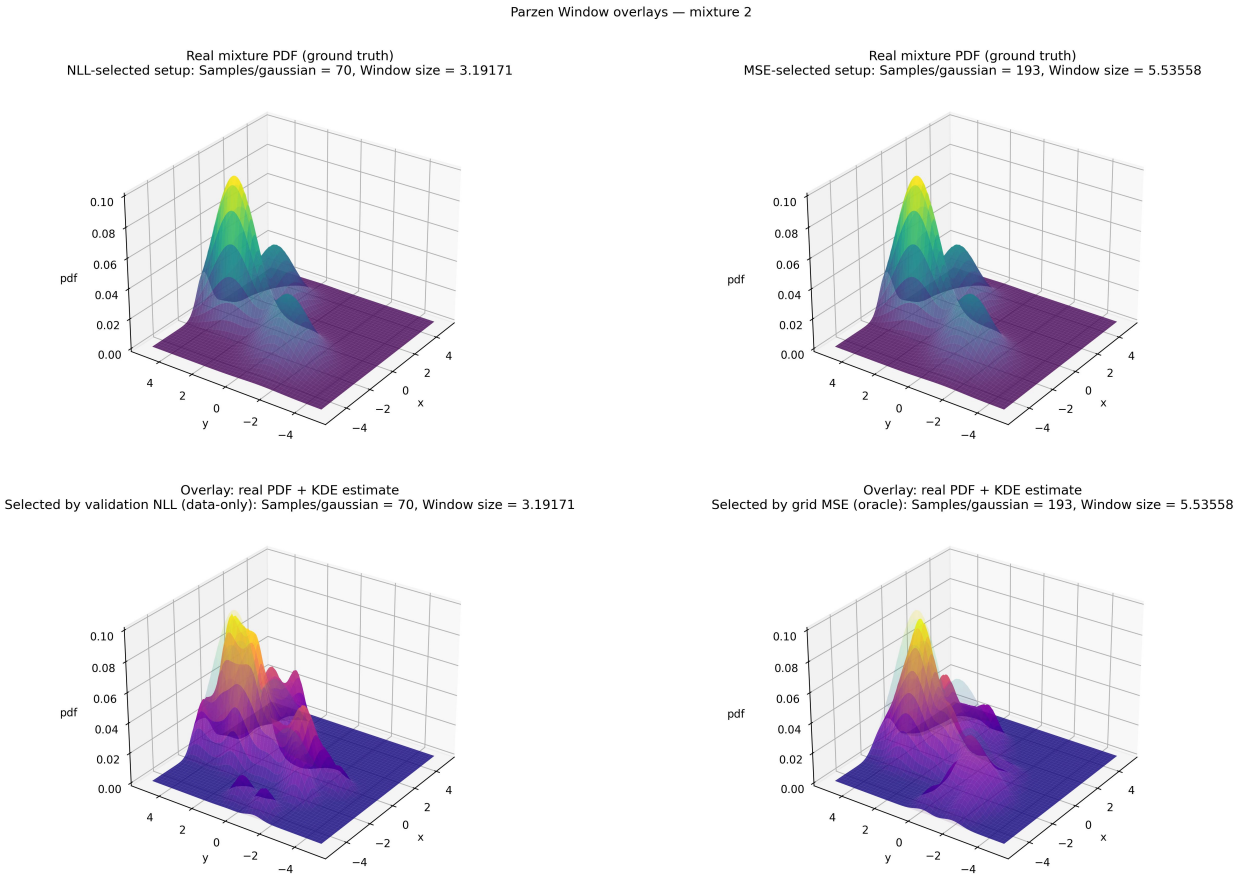

Figure 5: Top row: Mixture 2 pdf;
Bottom: PW estimate overlays selected by minimizing NLL and MSE respectively.

It's also noticeable how in this mixture that MSE selected parameters still cannot provide high accuracy in estimating low variance modes that are close to high variance ones,
this causes NLL selected overlays to better approximate these regions while still causing overhoots to the actual spikes and overall higher variance in the estimate; causing a lower MSE.

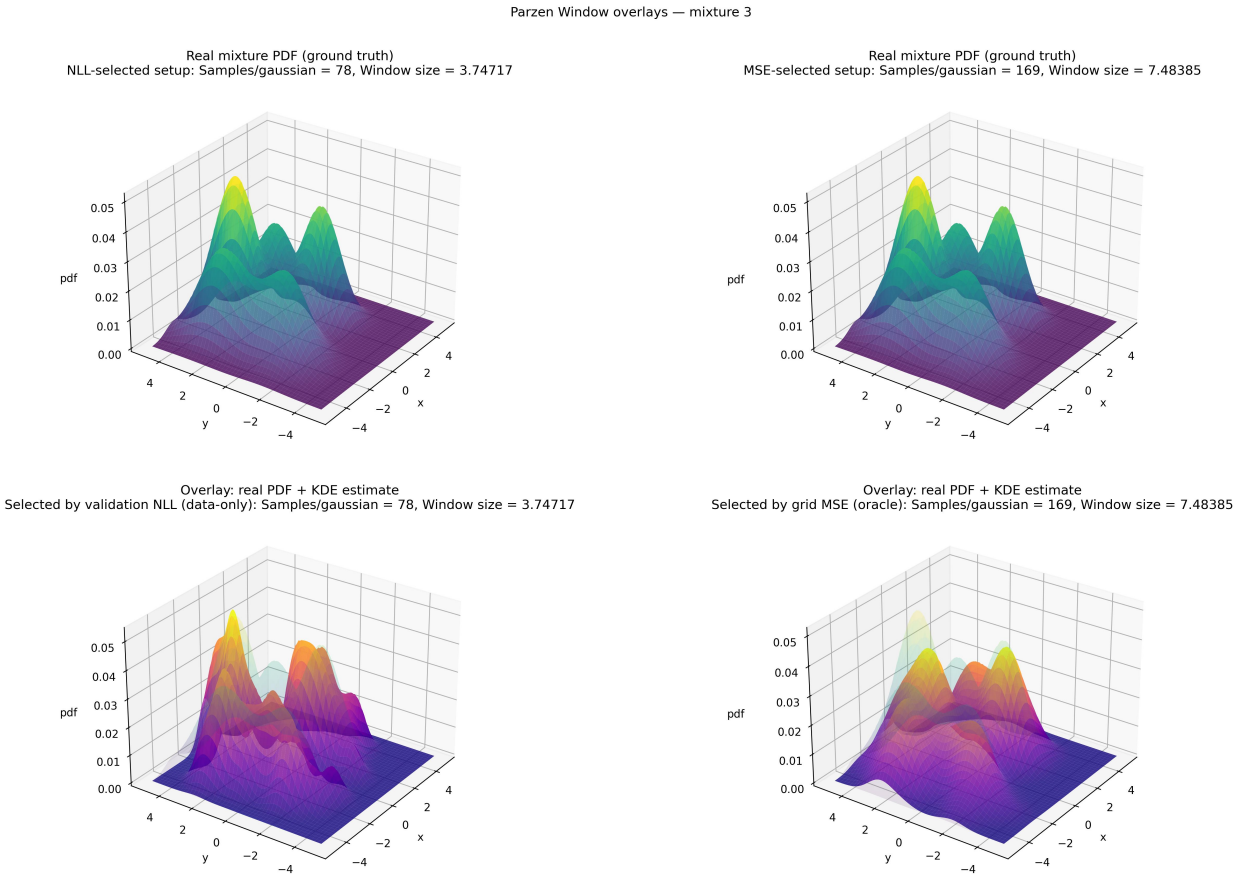

Figure 6: Top row: Mixture 3 pdf;
Bottom: PW estimate overlays selected by minimizing NLL and MSE respectively.

It was noted that kernels narrower than sample spacing create isolated peaks failing to merge into unified modes, generating spurious extra peaks. When kernel width falls below inter-sample distances, each kernel becomes independent, no longer contributing smoothly to adjacent peaks. MSE-optimized estimates cannot suppress this since the error metric only measures integrated error, allowing local over-sharpening at sample locations.

1.2 Parzen Neural Network

1.2.1 Parzen Neural Network Errors

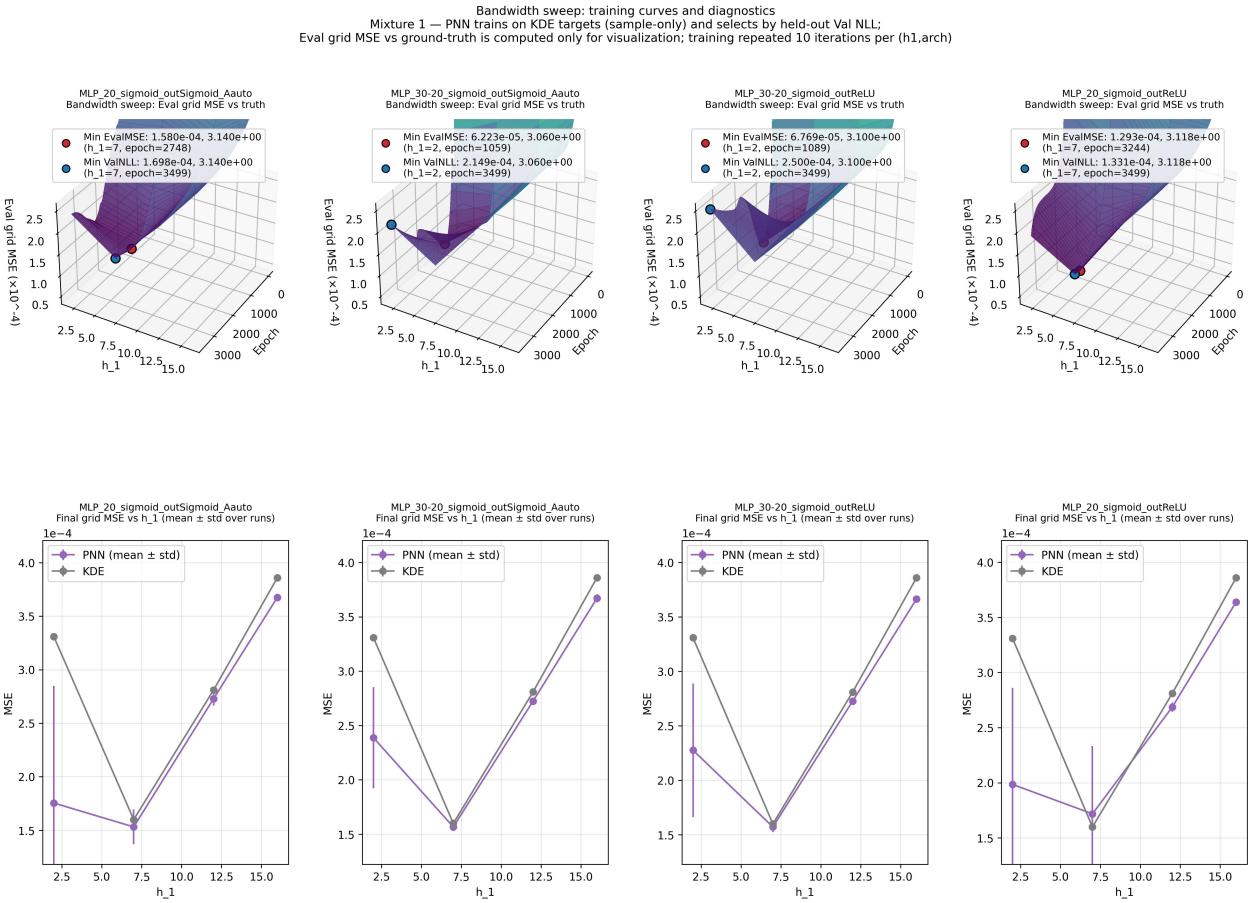

Figure 7: Top: Eval-MSE surface between mixture 1 pdf and PNN estimate over epoch × h_1
Bottom: final grid MSE (last epoch) vs h_1 shown as mean ± std
across 10 runs (PNN) and KDE for reference.

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.

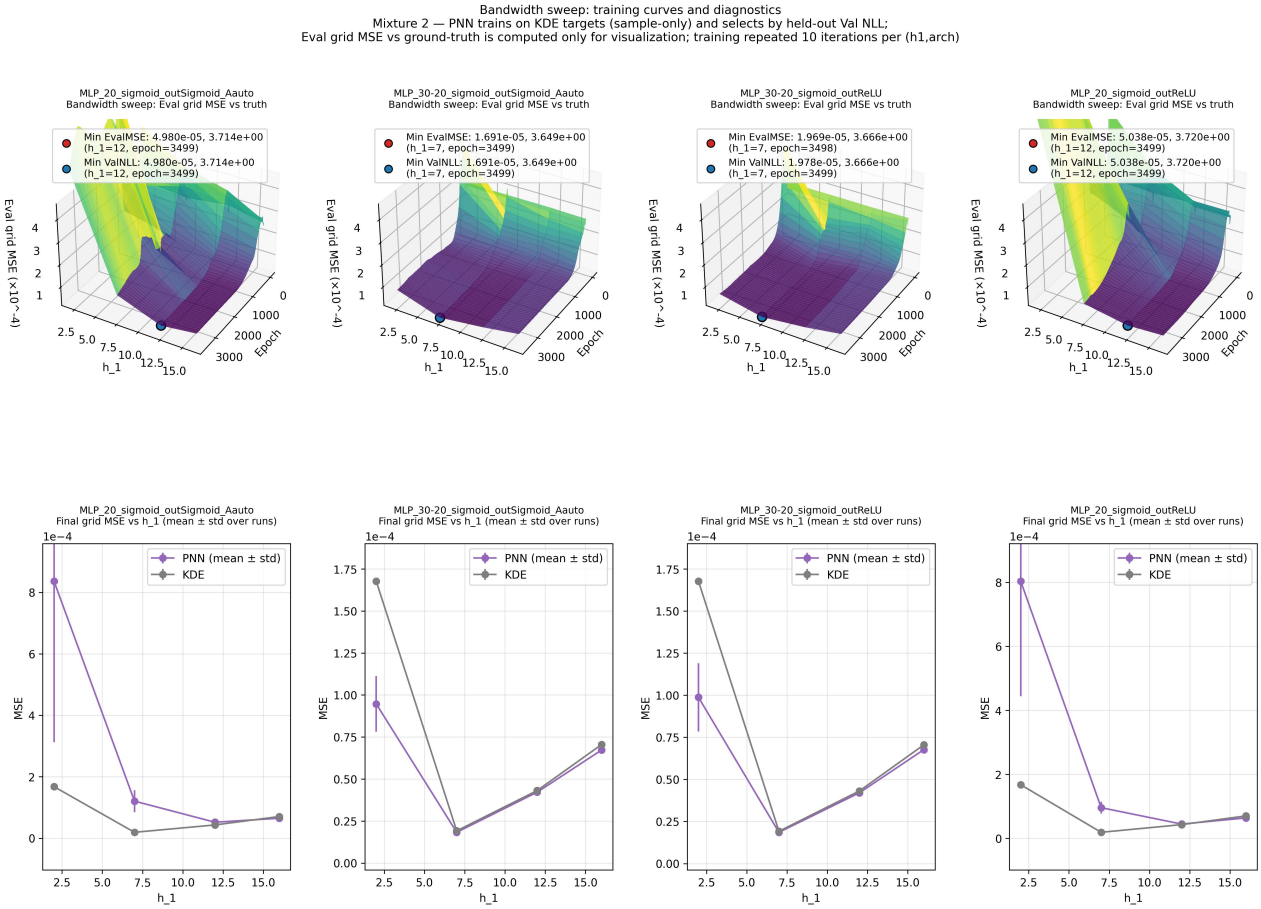

Figure 8: Top: Eval-MSE surface between mixture 2 pdf and PNN estimate over epoch $\times$ h_1
Bottom: final grid MSE (last epoch) vs h_1 shown as mean $\pm$ std
across 10 runs (PNN) and KDE for reference.

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

- It does not seem to make much difference for both deep and shallow architectures in the output layer is constructed with ReLus or Sigmoids with variable amplitude
- 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.

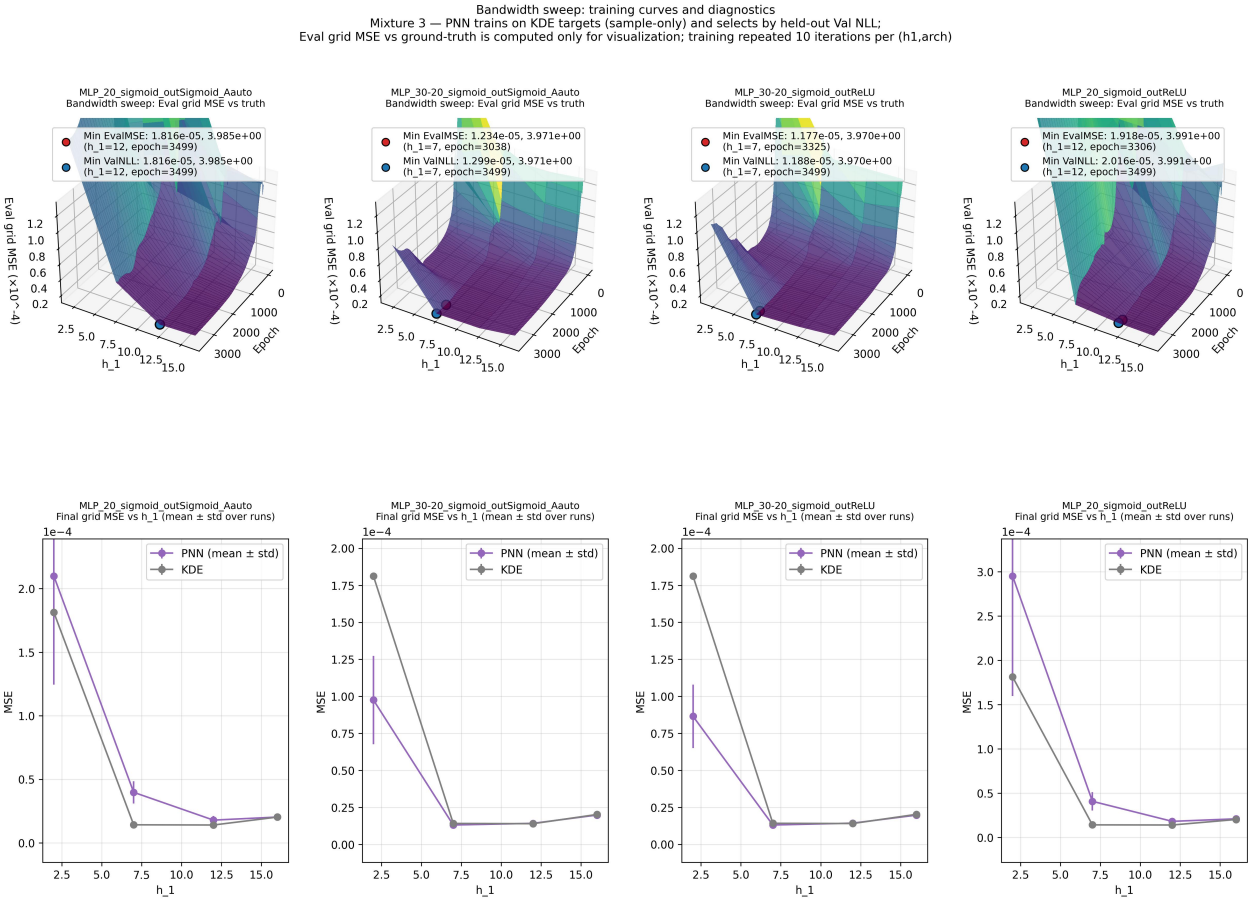

Figure 9: Top row: Eval-MSE surface between mixture 3 pdf and PNN estimate over epoch $\times$ h_1
Bottom row: final grid MSE (last epoch) vs h_1 shown as mean $\pm$ std
across 10 runs (PNN) and KDE for reference.

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).

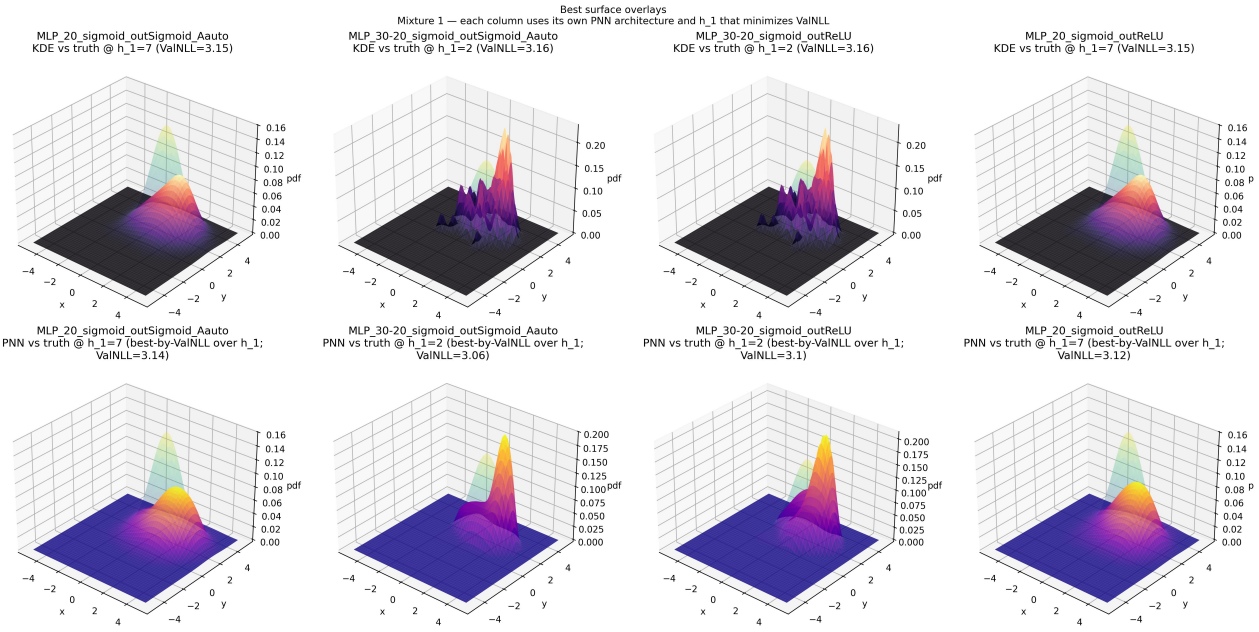

Figure 10: Best overlays for mixture 1, selected by minimizing ValNLL.
Top row: Parzen Window (KDE) estimate vs ground-truth mixture PDF at the same selected h_1.
Bottom row: Parzen Neural Network (PNN) estimate vs ground-truth mixture PDF.

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??

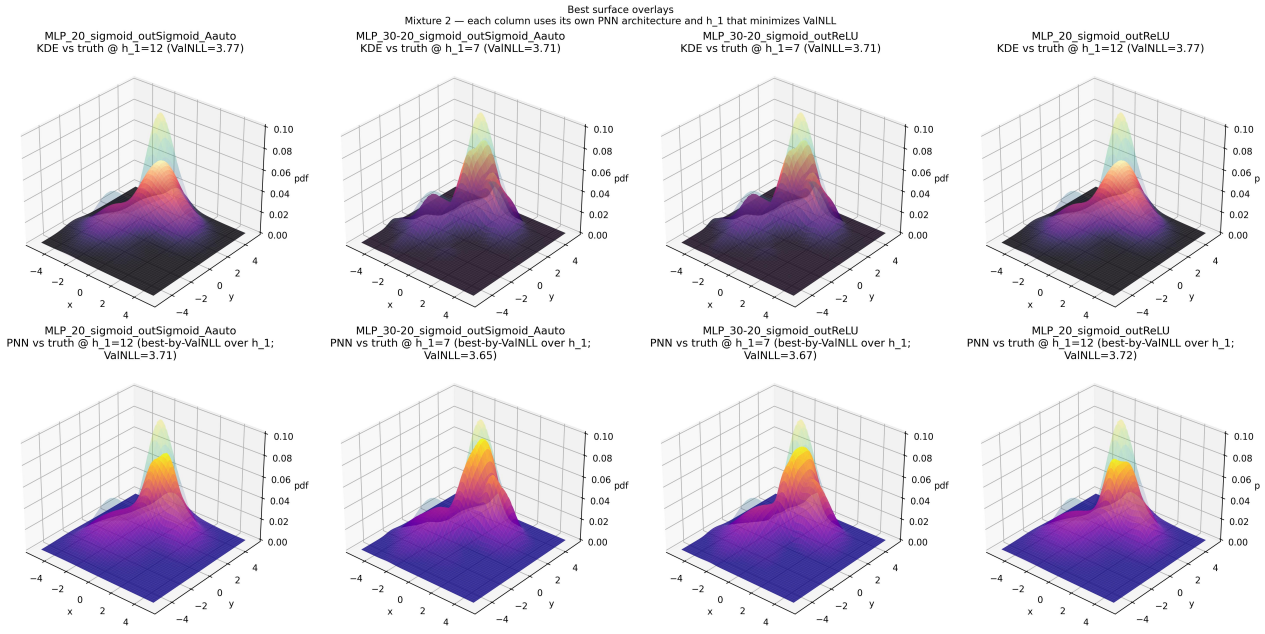

Figure 11: Best overlays for mixture 2, selected by minimizing ValNLL.
Top row: Parzen Window (KDE) estimate vs ground-truth mixture PDF at the same selected h_1.
Bottom row: Parzen Neural Network (PNN) estimate vs ground-truth mixture PDF.

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??.

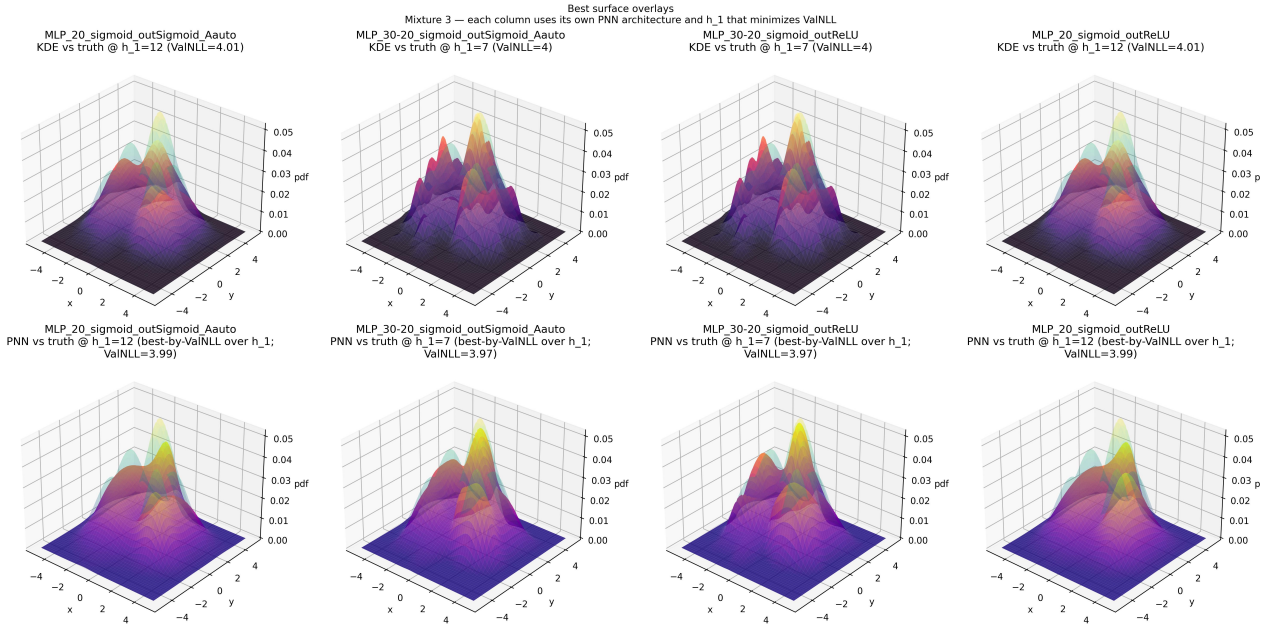

Figure 12: Best overlays for mixture 3, selected by minimizing ValNLL.
Top row: Parzen Window (KDE) estimate vs ground-truth mixture PDF at the same selected h_1.
Bottom row: Parzen Neural Network (PNN) estimate vs ground-truth mixture PDF.

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.

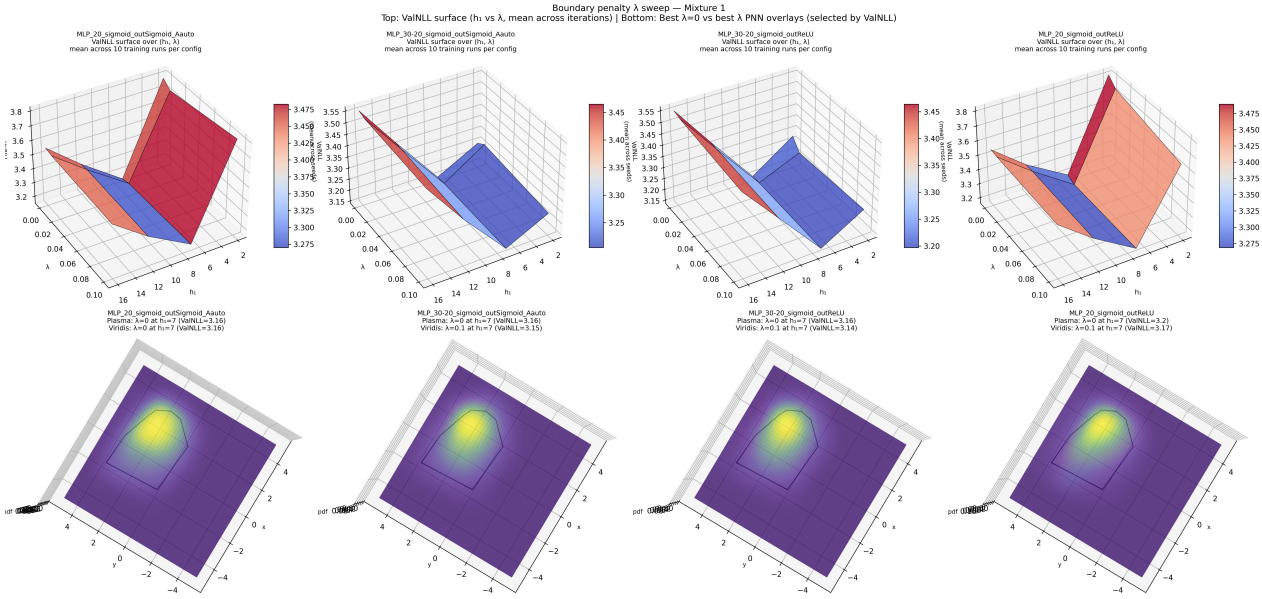

Figure 13: Top Row: ValNLL for mixture 1 vs $h_1 \times \lambda$ for every PNN architecture
Bottom Row: PNN overlay that minimizes ValNLL@$\lambda = 0$
vs the PNN that minimizes ValNLL $\forall\lambda$

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.

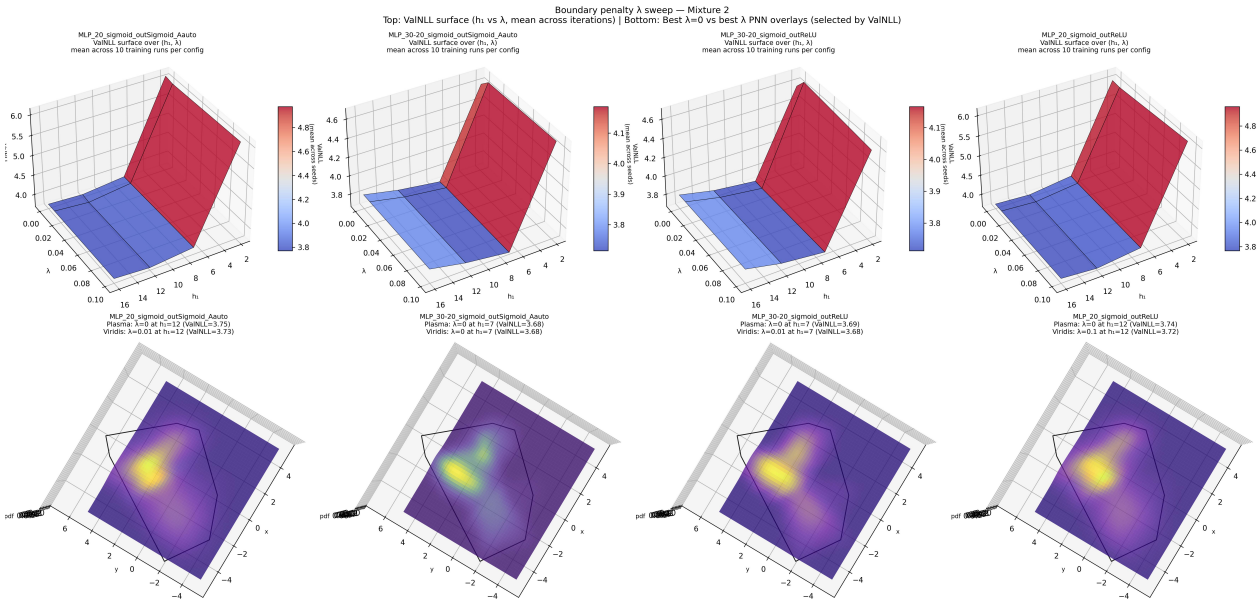

Figure 14: Top Row: ValNLL for mixture 2 vs $h_1 \times \lambda$ for every PNN architecture
Bottom Row: PNN overlay that minimizes ValNLL@$\lambda = 0$
vs the PNN that minimizes ValNLL $\forall\lambda$

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????

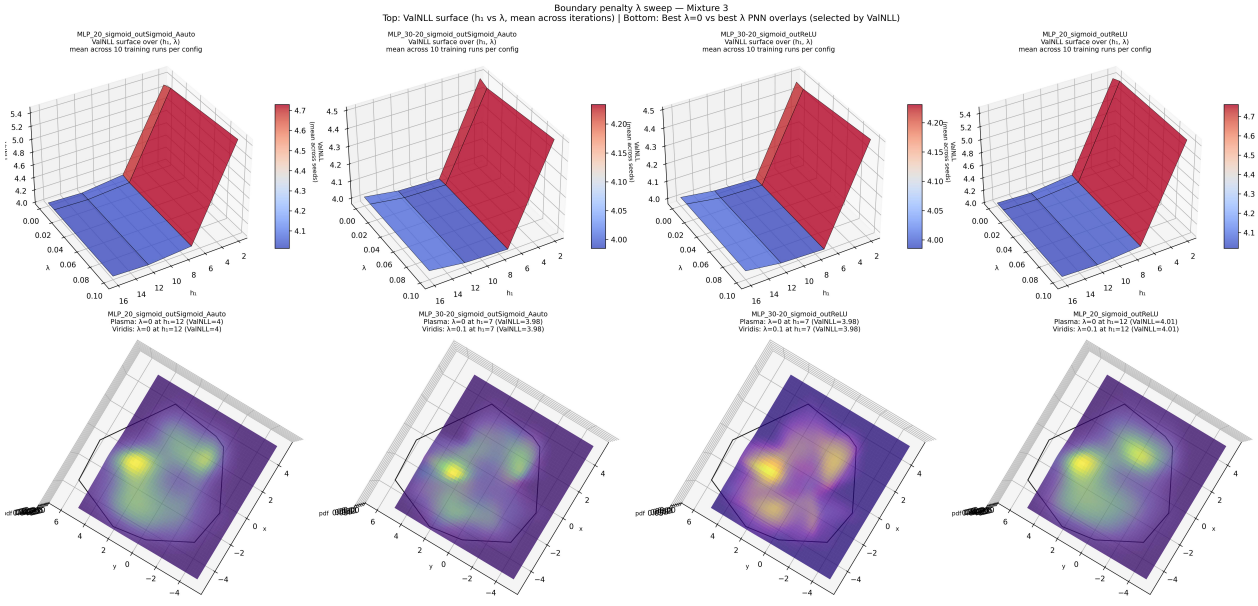

Figure 15: Top Row: ValNLL for mixture 3 vs $h_1 \times \lambda$ for every PNN architecture
Bottom Row: PNN overlay that minimizes ValNLL@$\lambda = 0$
vs the PNN that minimizes ValNLL $\forall\lambda$

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.