# V1 data

Code: [v1\_comparison\_v2.m](https://github.com/weigcdsb/COM_POISSON/blob/main/demo/v1/v1_comparison_v2.m). Neuron = 13

**5 models are fitted:**

1. CMP-LDS: nknots = 5 for lambda, =3 for nu
2. CMP-LDS: nknots = 5 for lambda, =1 for nu
3. Poisson-LDS: nknots = 5 for lambda
4. Static CMP: nknots = 5 for lambda, = 3 for nu
5. Static Poisson: nknots = 5 for lambda

Model (1), (2) and (3) are estimated by smoother + Newton-Raphson. So, they are MAP’s now. The Q are estimated in the simplified version: diag([a10, a11,…, a11 ]) for lambda and diag([a20, a21,…, a21 ]) for nu.

For the “window” method, I now think we should give it up. For the mean estimate, I can (conceptually) show the asymptotic unbiasedness. However, it will be a disaster for inference: the variance is underestimated. This is why when estimating Q, we should never turn on “window”.

I didn’t include the CMP-LDS model with constant nu, since it’s quite tricky (and what I did before was wrong!). When nu is constant but unknown, we need to see all the data to estimate the single nu. This breaks the filtering/ Markovian structure. Or we can think it another way: the hessian is no longer tri-block diagonal, the nu is correlated to {beta1, beta2, …, betaT}. Therefore, it’s impossible to do newton-Raphson efficiently now…

Maybe we can first integrate {beta1, beta2, …, betaT} out and get the MAP of nu, and then get MAP for {beta1, beta2, …, betaT} by profile posterior. But that makes things too complicated, and I think it’s not the main story we want to talk.

**Training & held-out data set**

In each trial, half data is randomly chosen for training. (So, the training data is “speckled”). The full parameter trace is recovered by interpolation:

function theta\_ho = theta\_interp1(theta\_tr, trIdx, hoIdx)

theta\_ho = zeros(size(theta\_tr,1), length(hoIdx));

for d = 1:size(theta\_tr,1)

theta\_ho(d,:) = interp1(trIdx,theta\_tr(d,:),hoIdx);

end

if(isnan(theta\_ho(:,end))); theta\_ho(:,end) = theta\_tr(:,end);end

end

The log-likelihood per spike

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | CMP-LDS(5,3) | CMP-LDS(5,1) | Poisson-LDS(5) | Static-CMP(5,3) | Static-Poisson(5) |
| Training llhd/spike | -0.9101 | -0.9108 | -0.9307 | -0.9266 | -0.9507 |
| Held-out llhd/spike | -0.9099 | -0.9099 | -0.9338 | -0.9189 | -0.9452 |

The CMP-LDS(5,3) and CMP-LDS(5,1) are close to each other. (In terms of held-out llhd, sometimes CMP-LDS(5,3) wins and sometimes CMP-LDS(5,1) wins in these limited fitting). But using CMP is very important! Even the static CMP beats the Poisson-LDS (always in limited examples). Turning on the non-stationarity of parameters in CMP improves the fitting a bit.

Plots (caution: scales for obs. and fitted are different)

CMP-LDS(5,3)



CMP-LDS(5,1)



# Hippocampus data

Code: [hc\_comparison\_v2.m](https://github.com/weigcdsb/COM_POISSON/blob/main/demo/hc/hc_comparison_v2.m). Neuron = 12

Nknots = 10 for lambda, nknots = 1 for nu.

