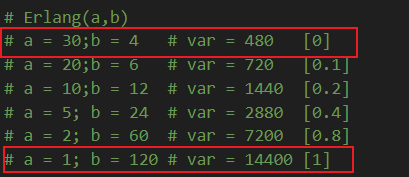
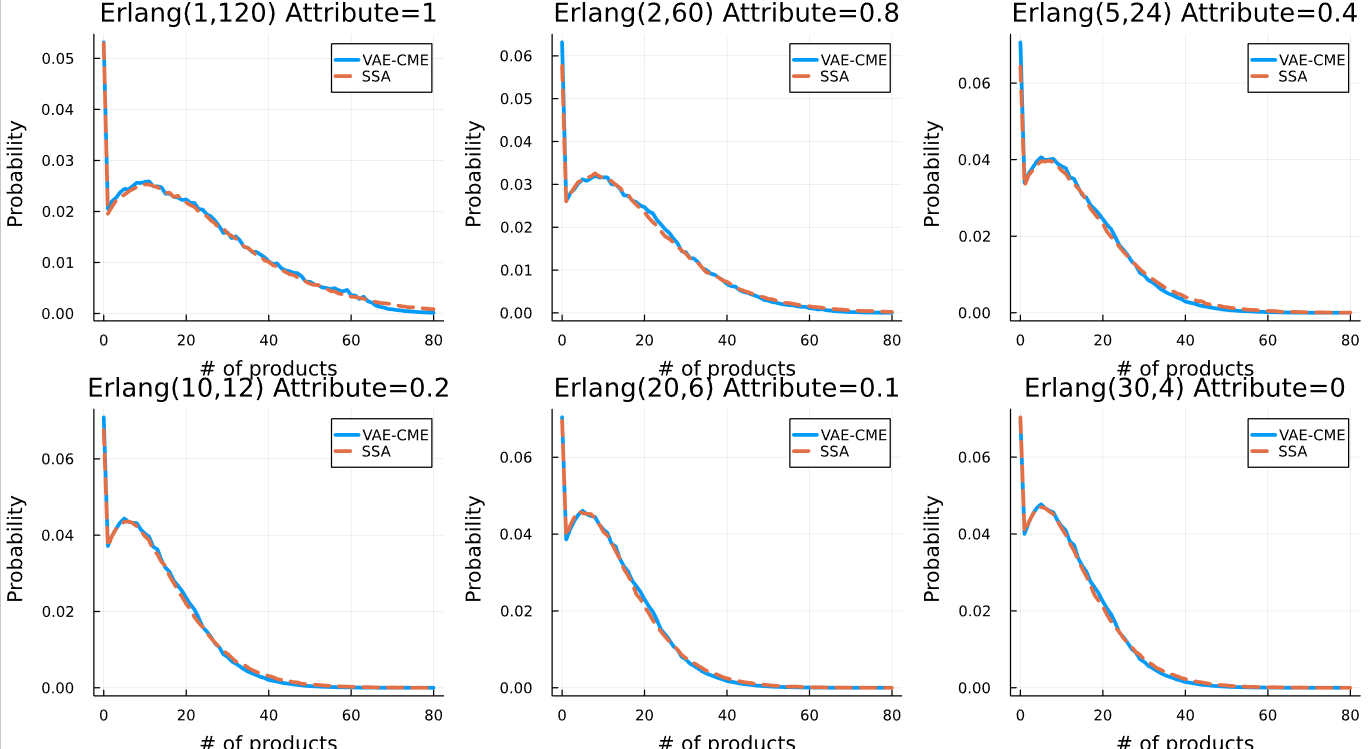


按照nc的取值

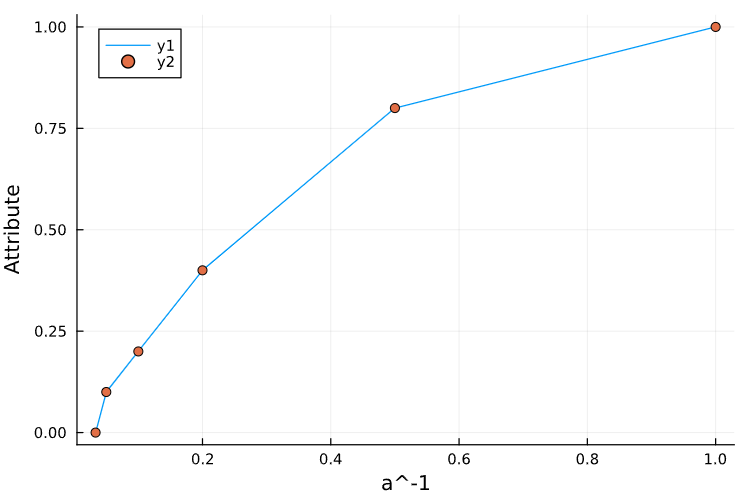
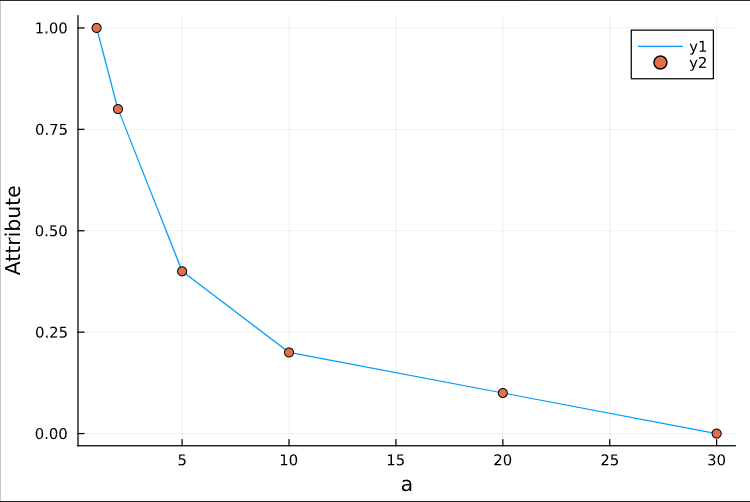
先用Erlang分布做实验，先确定Attribute的线性关系，\alpha = 0.0282，\beta = 3.46

a 最小取1，30往上的话SSA分布的差距就变得很小了

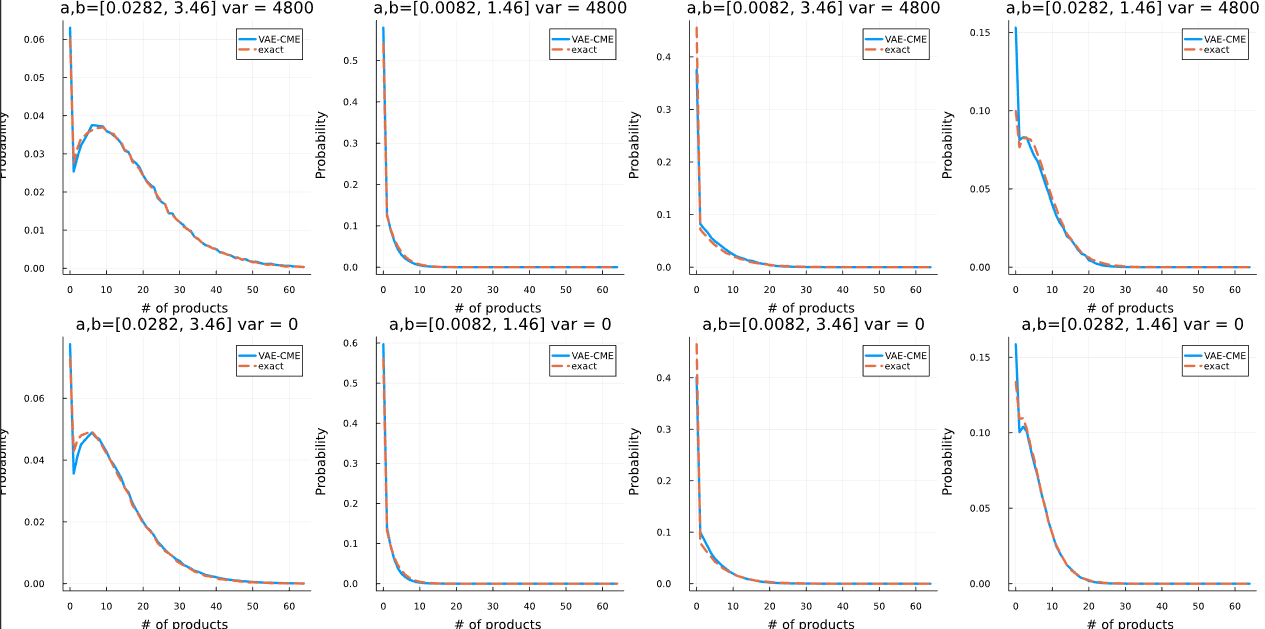




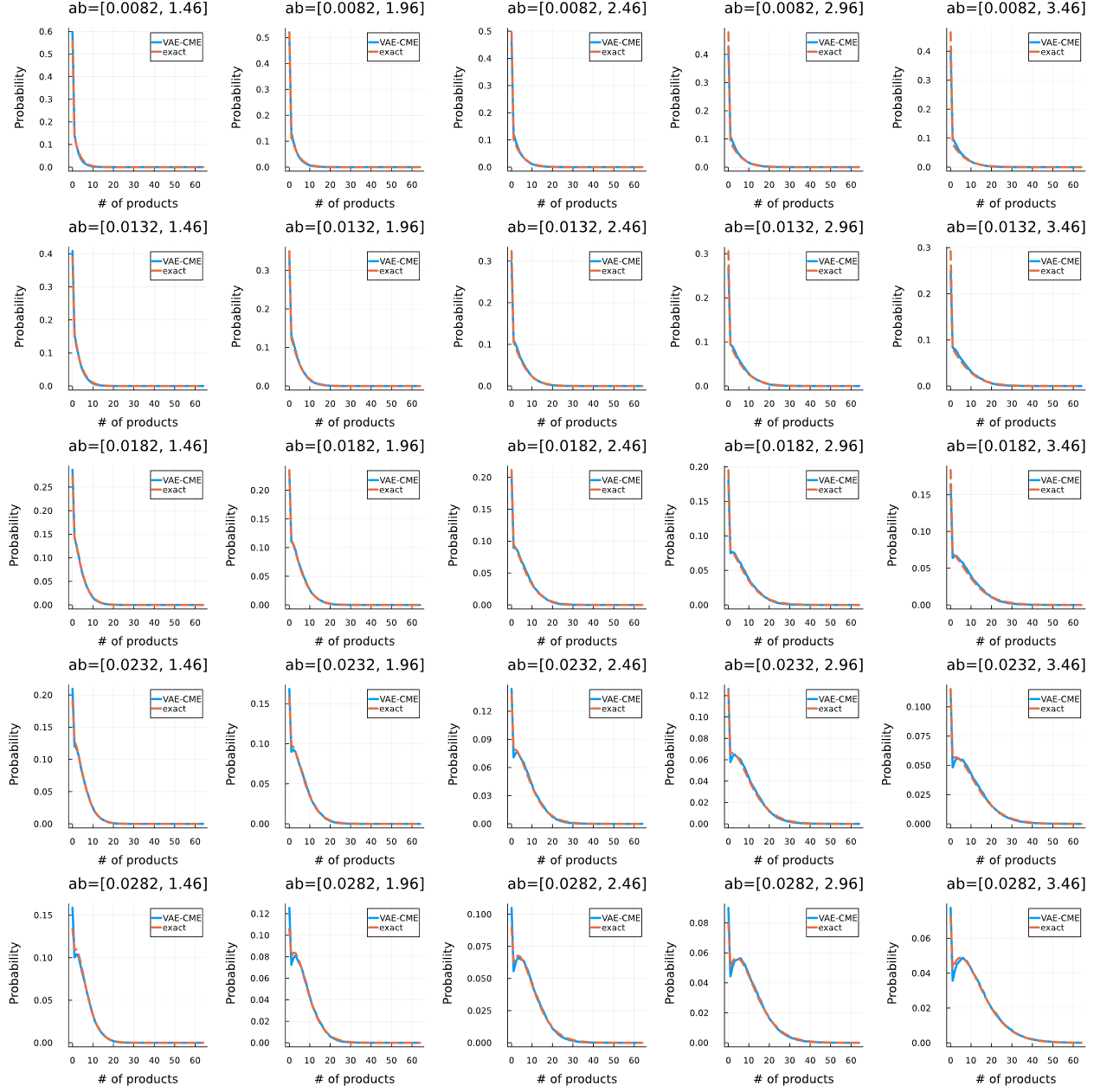
难以找出其线性关系诶

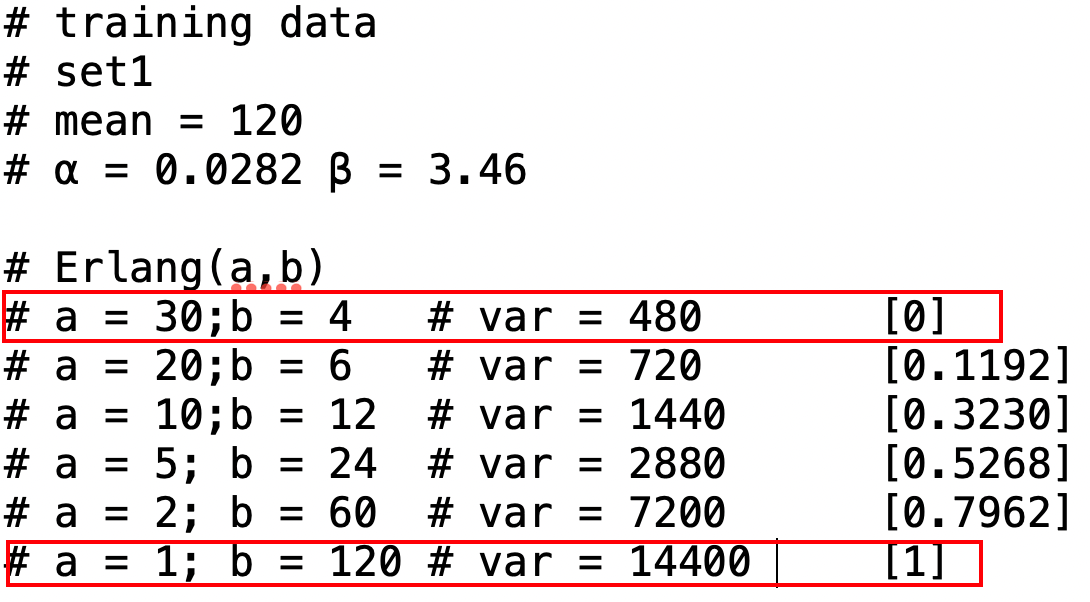


另外，增加数据的想法，似乎不太可行

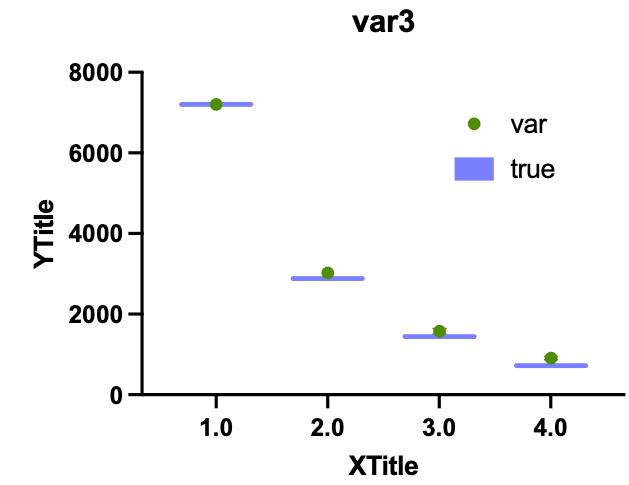


拟合效果不尽人意

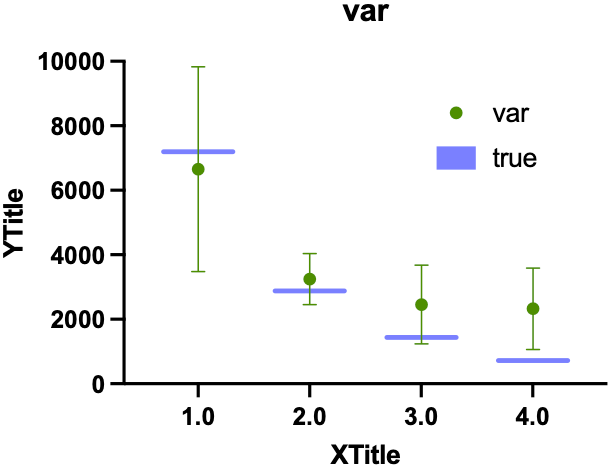
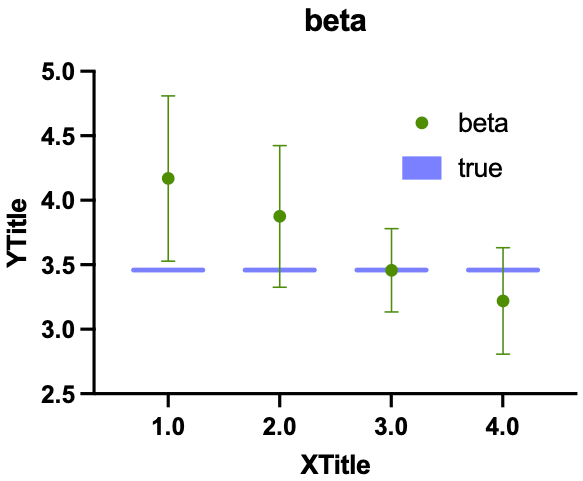
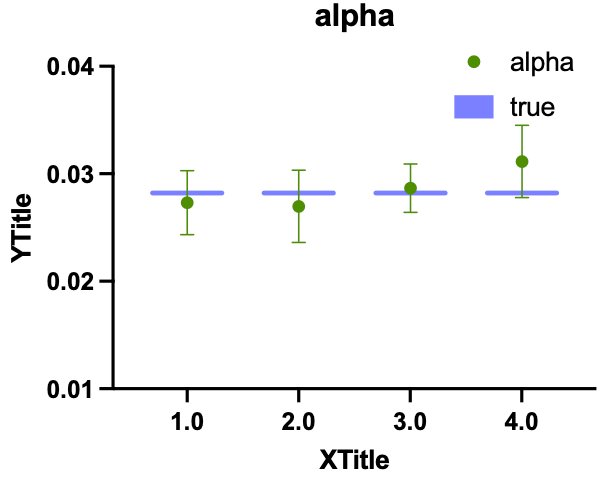




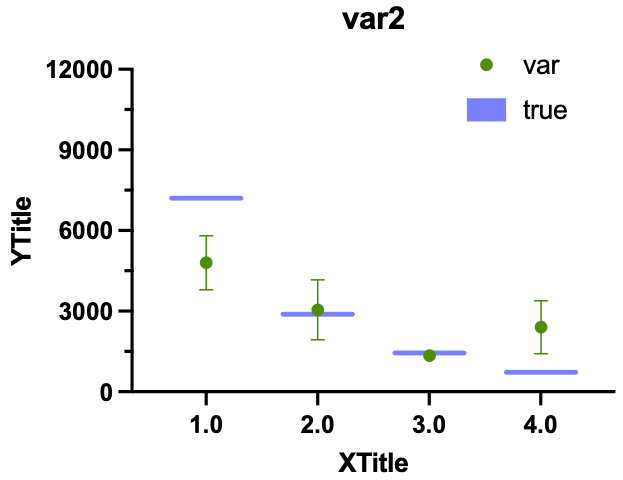
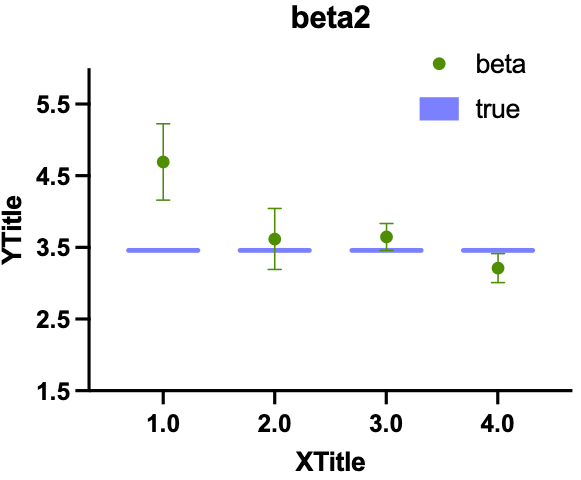
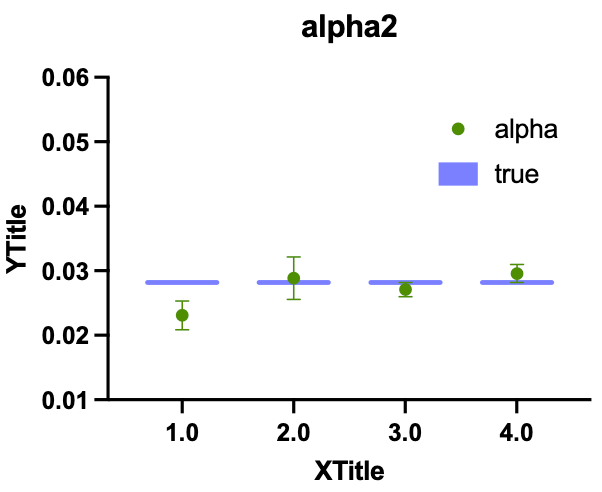
对alpha和beta为固定值的情况进行了Inference的测试，红色框框为训练集，中间4组为做Inference的数据，数据量是5000次SSA，在Inference中如果仅仅预测方差一个变量，效果还可以



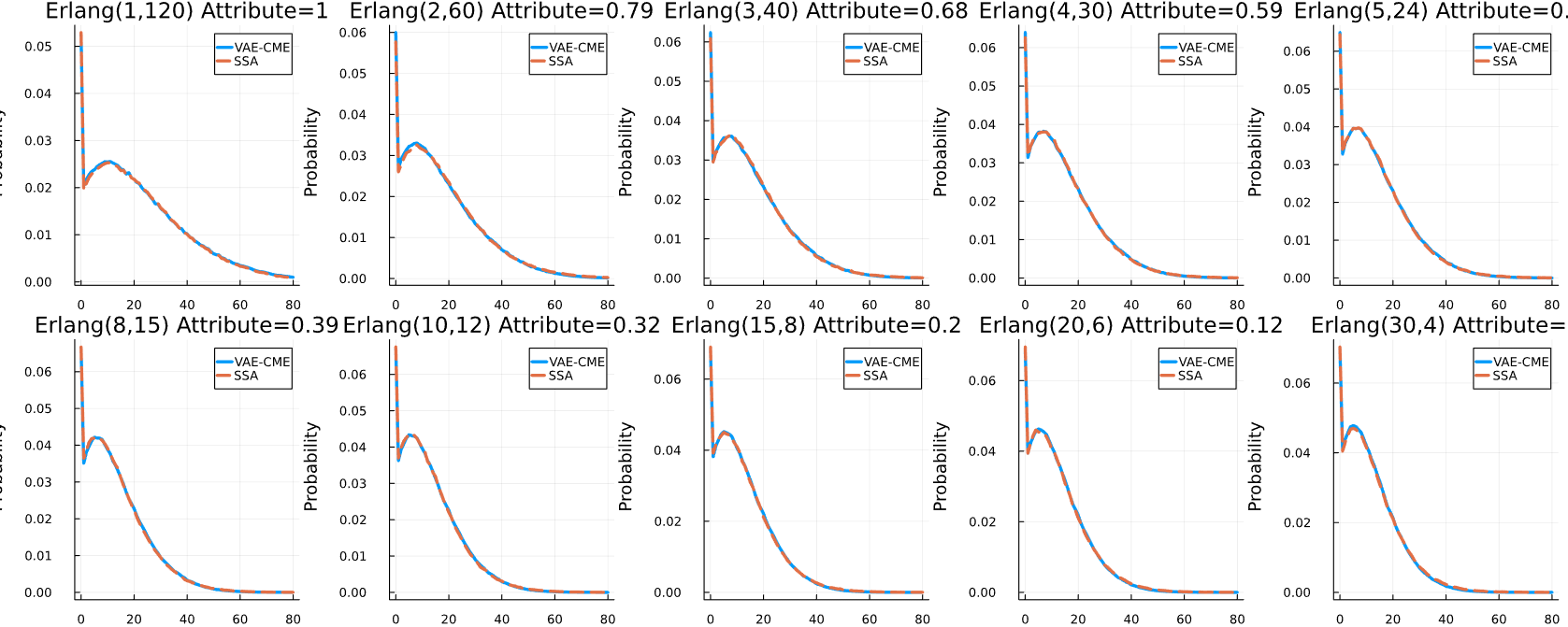
但如果预测alpha beta var三个值的时候，效果感觉差点意思



所有初始值设置为真值时的Inference效果



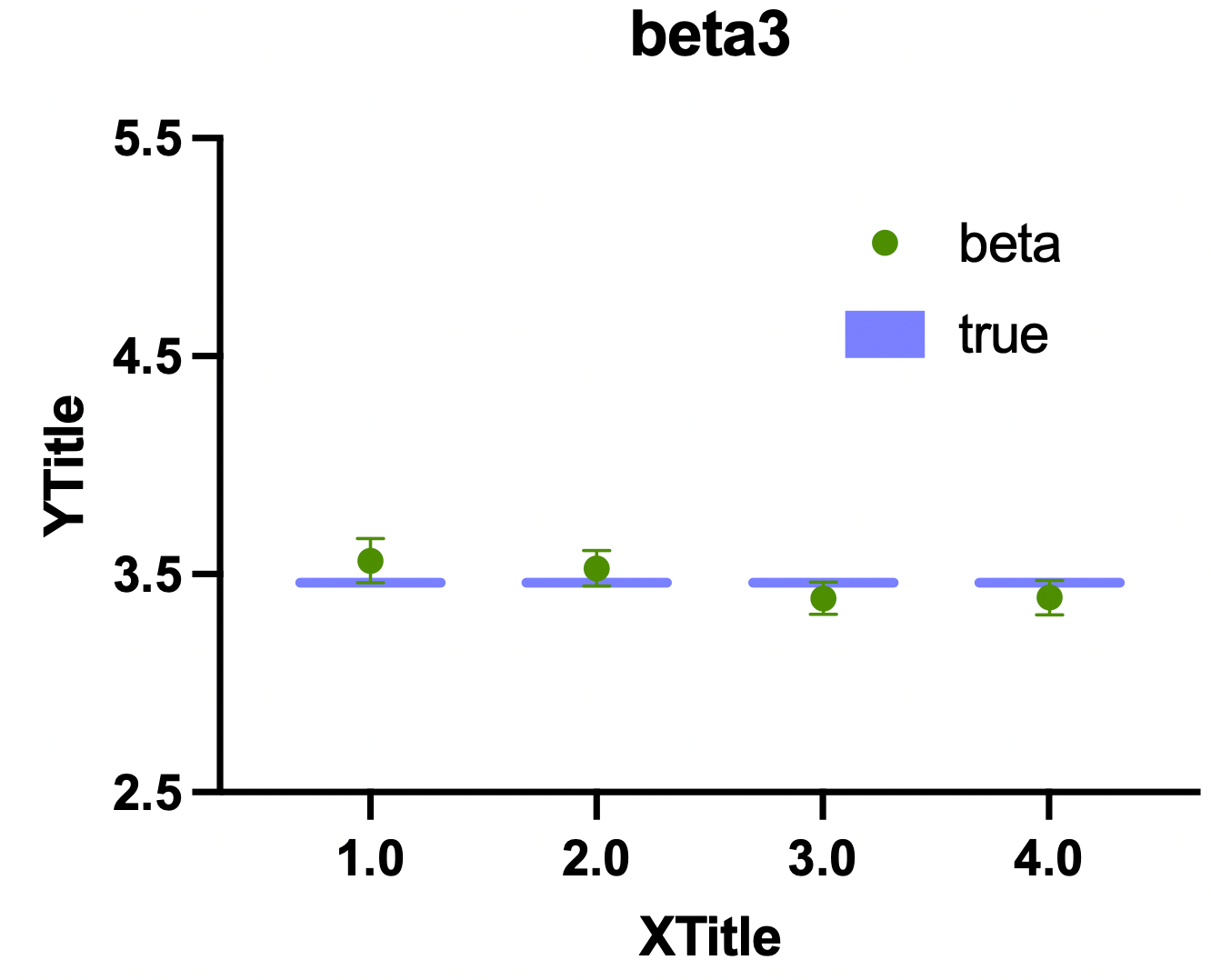
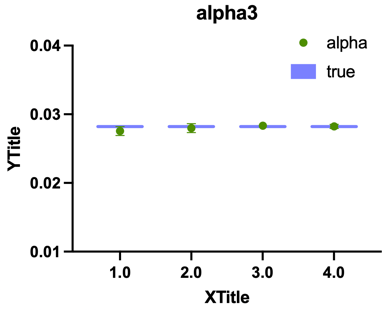
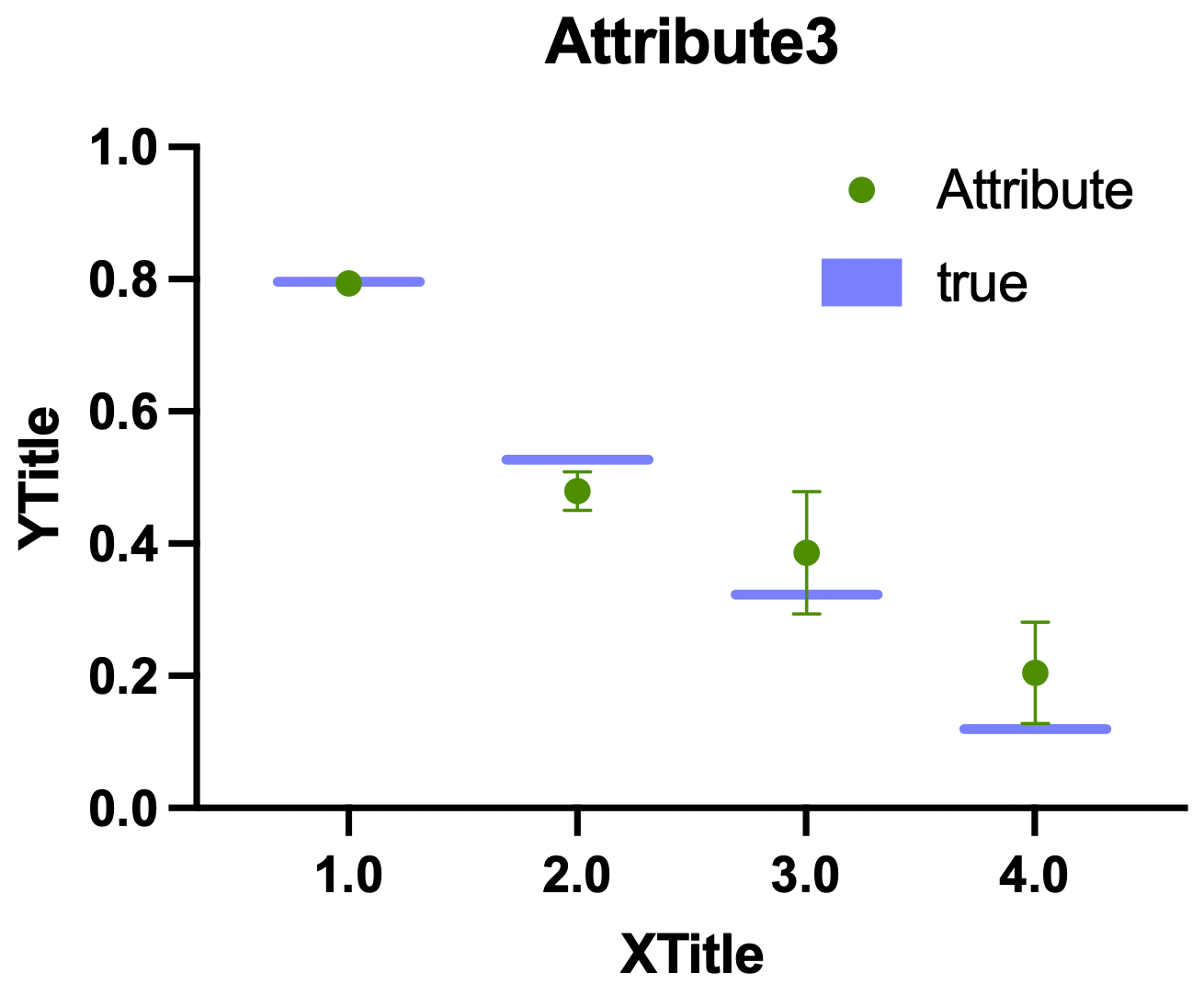
增加了一组，erlang(5,24) Attribute = 0.52，训完之后按照log的线性关系，预测几乎严丝合缝，说明确实是log的线性关系，此时alpha=0.0282，beta=3.46，只有一组alpha和beta



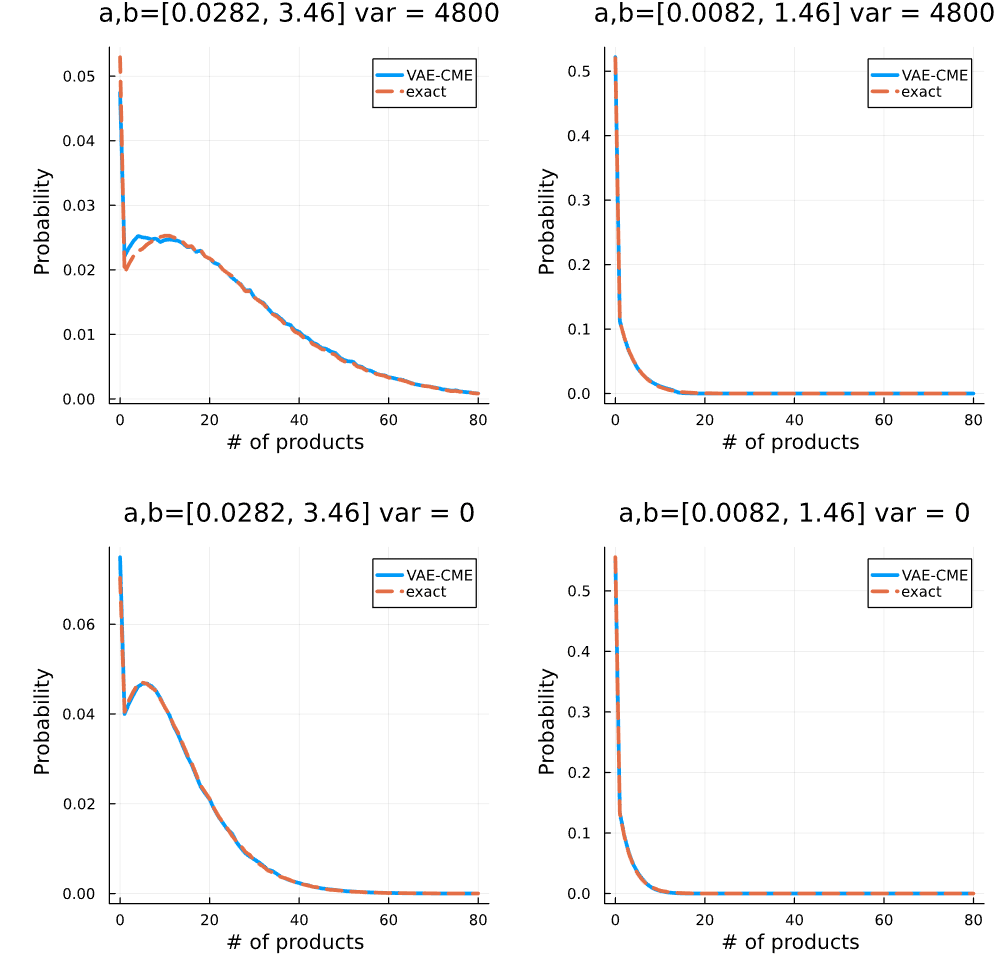
用50w次SSA对Erlang(10,12)进行预测，同一组SSA，重复五次实验，不同的初值

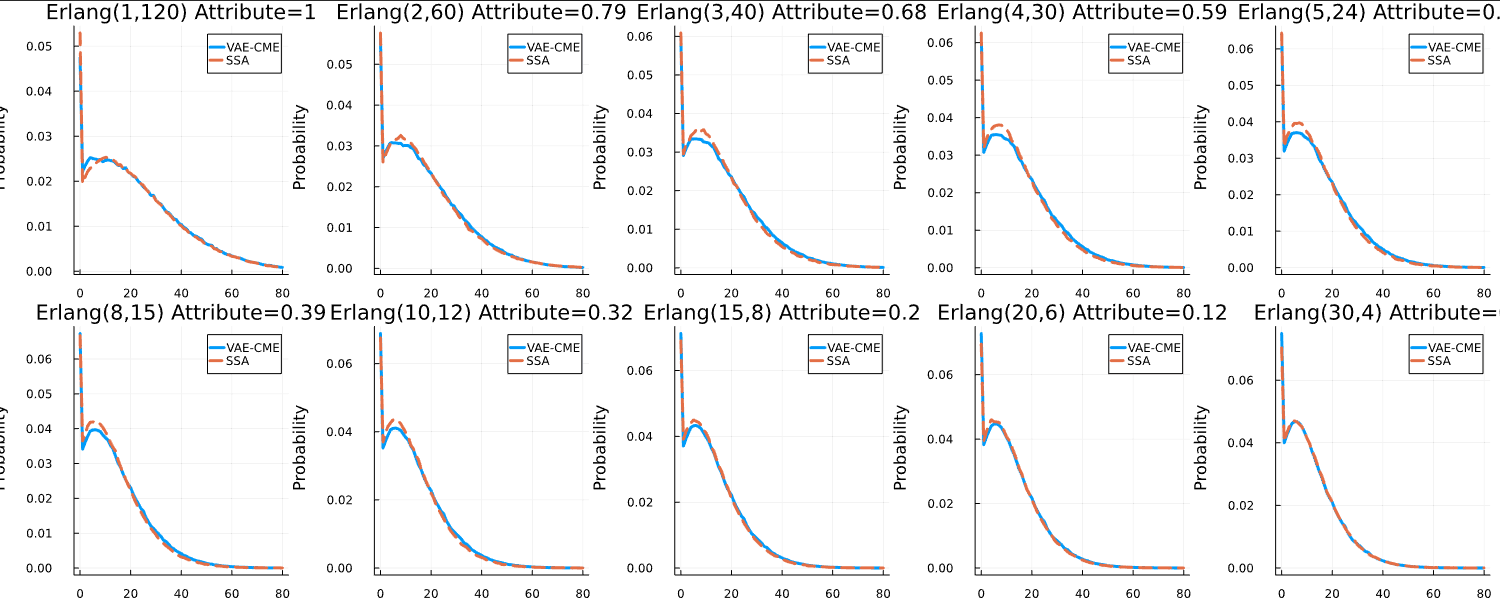


设置初值为真值，每一组参数，每一组参数只有tau的var不同，同样的参数跑出五组5000次的SSA，对五组不同SSA数据进行Inference

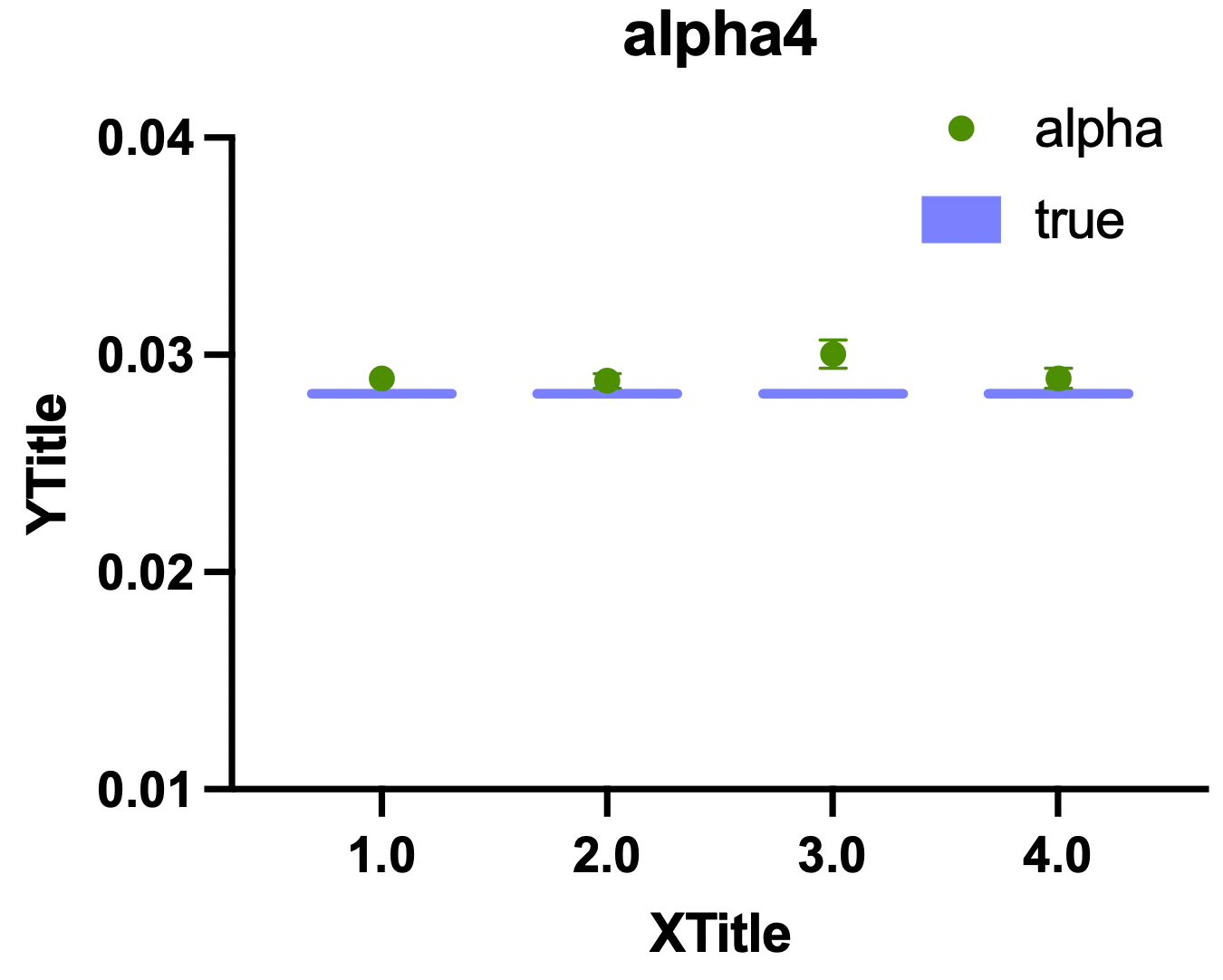
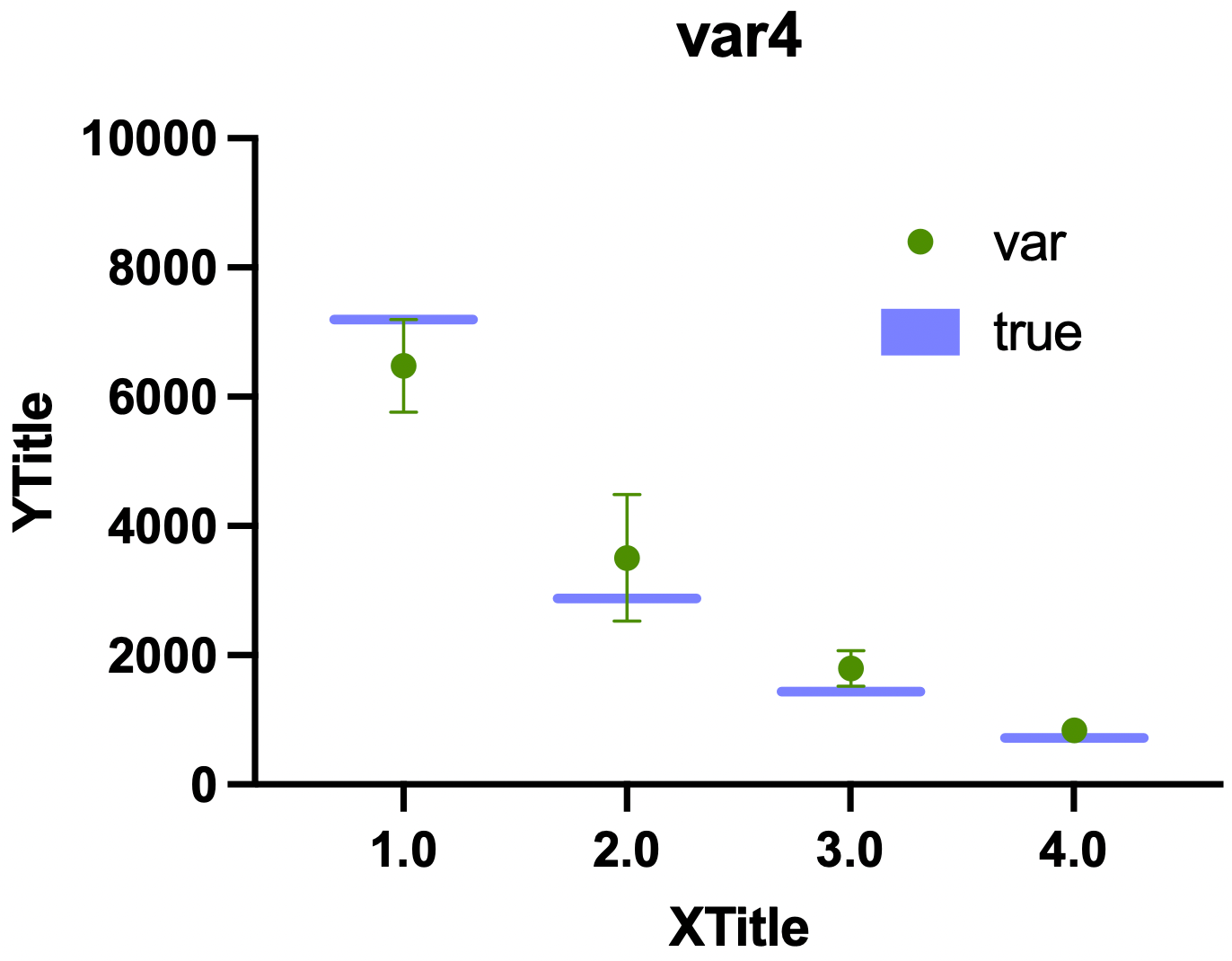
 

增加一组的alpha和beta的情况，数据是Attribute=0和1，拟合和预测效果

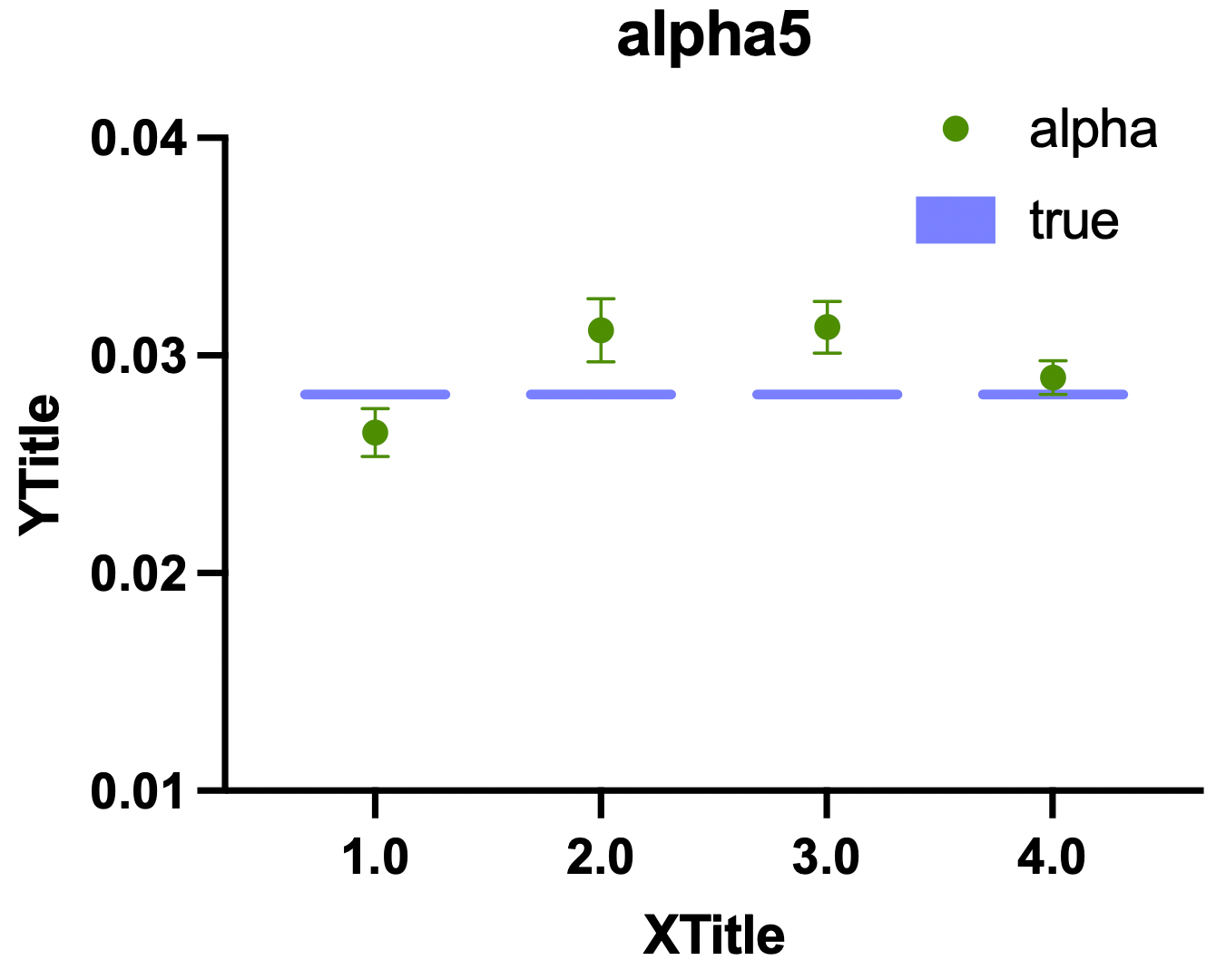
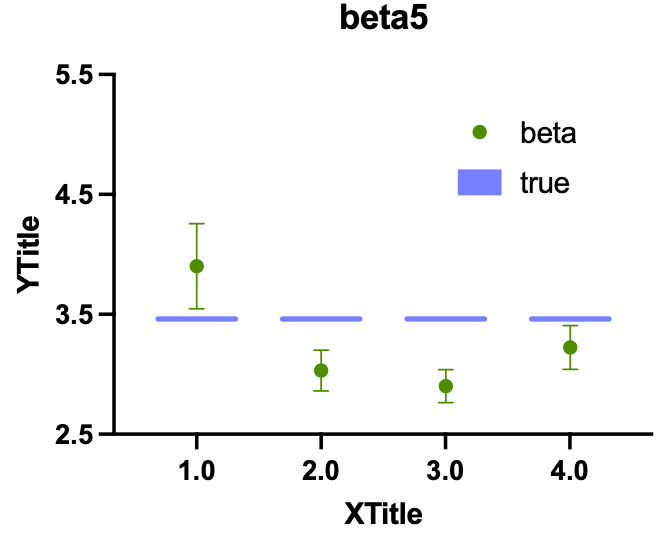
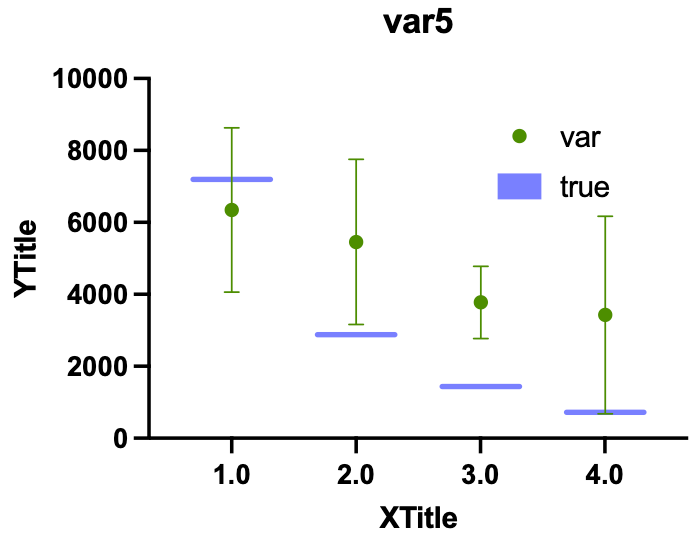




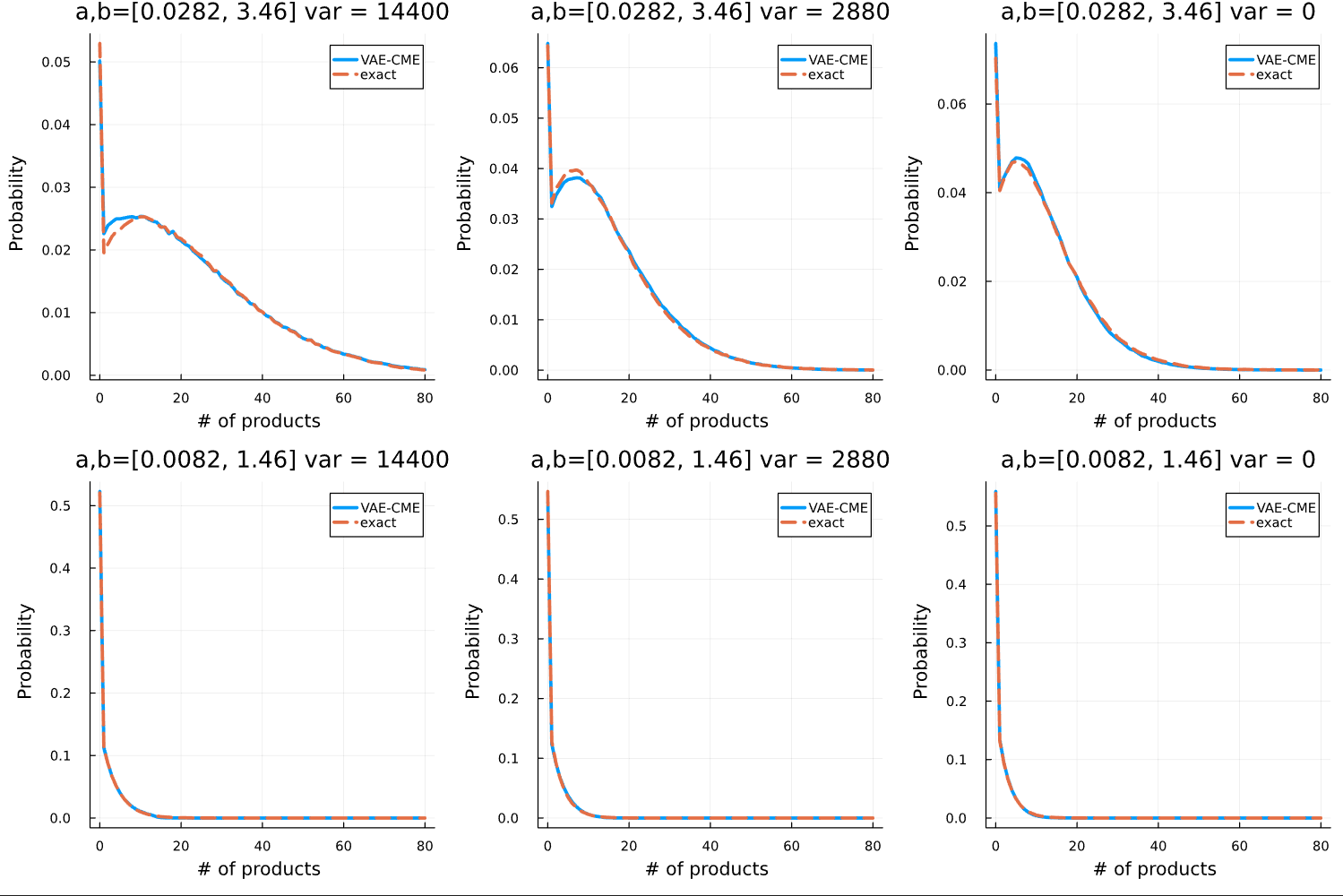
对alpha=0.0282，beta=3.46的Inference效果，50w次SSA

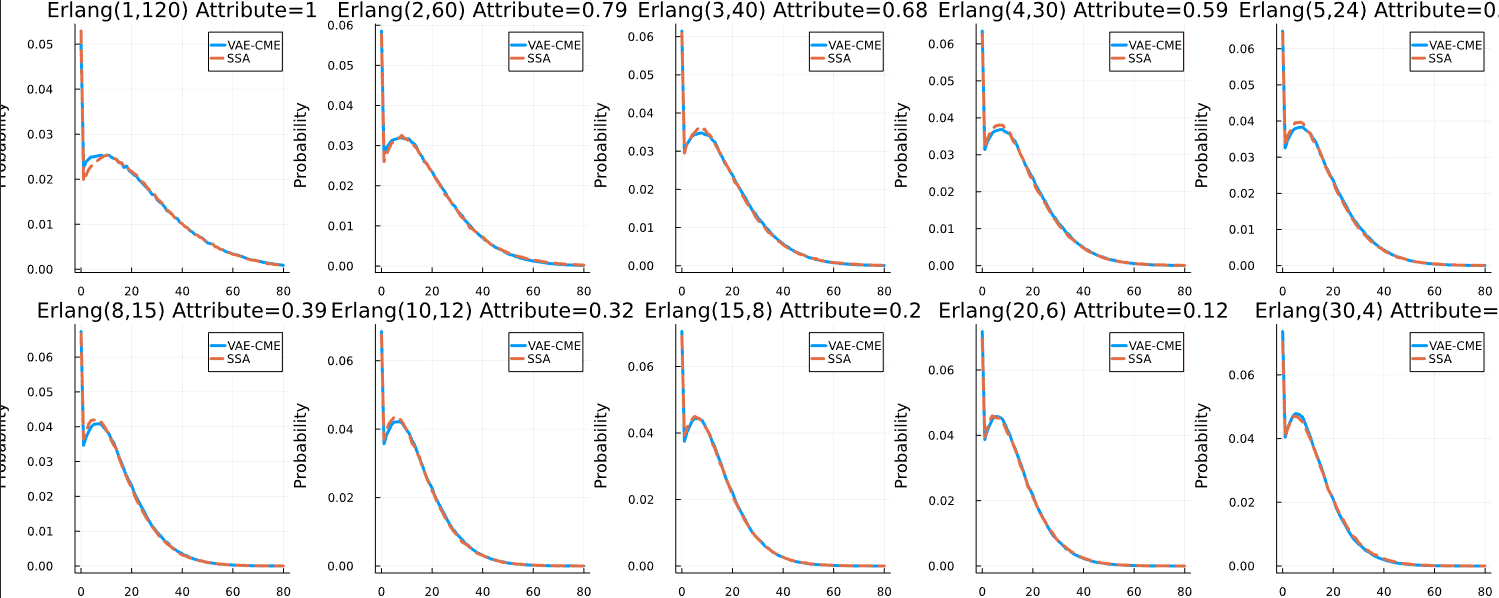
  

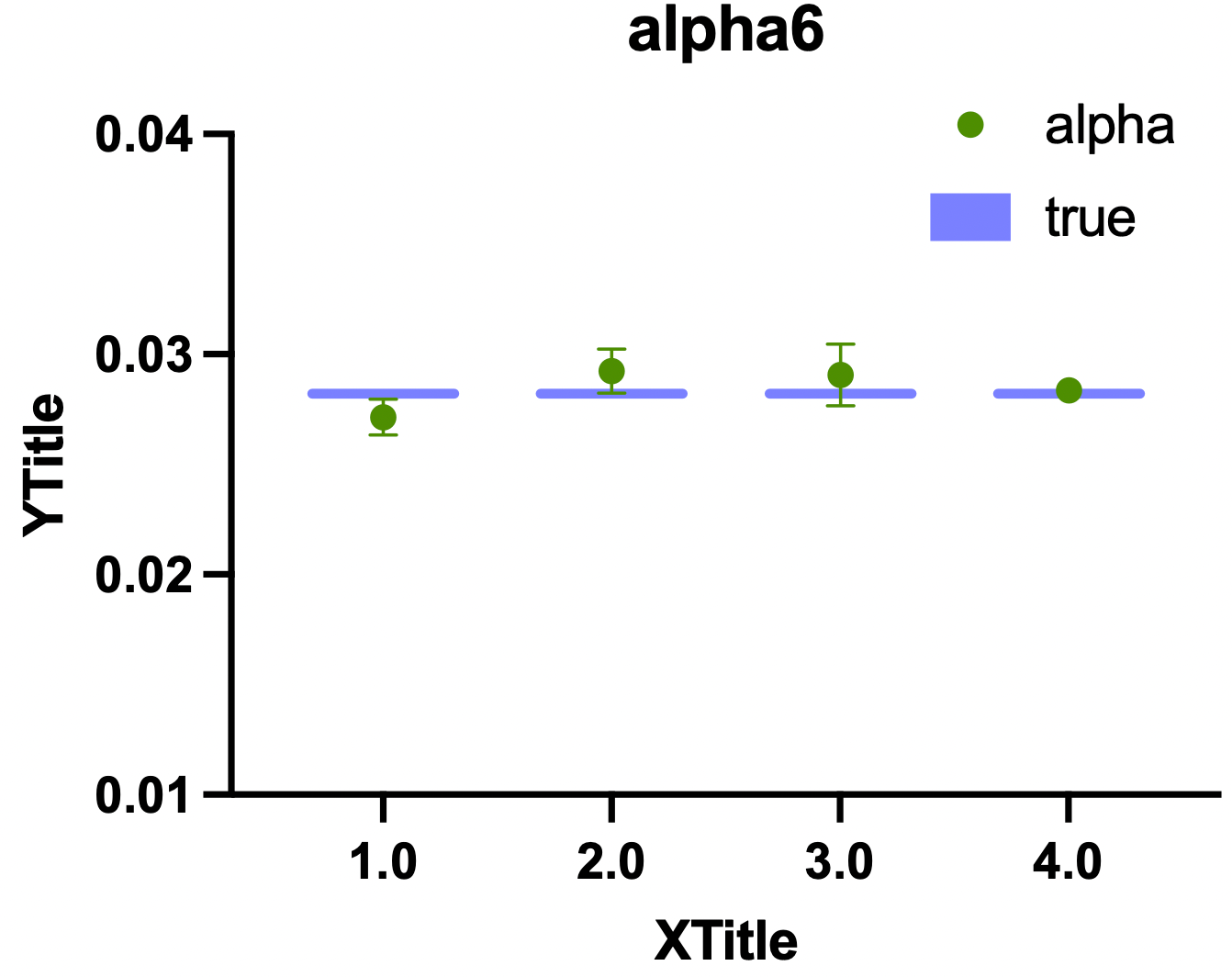
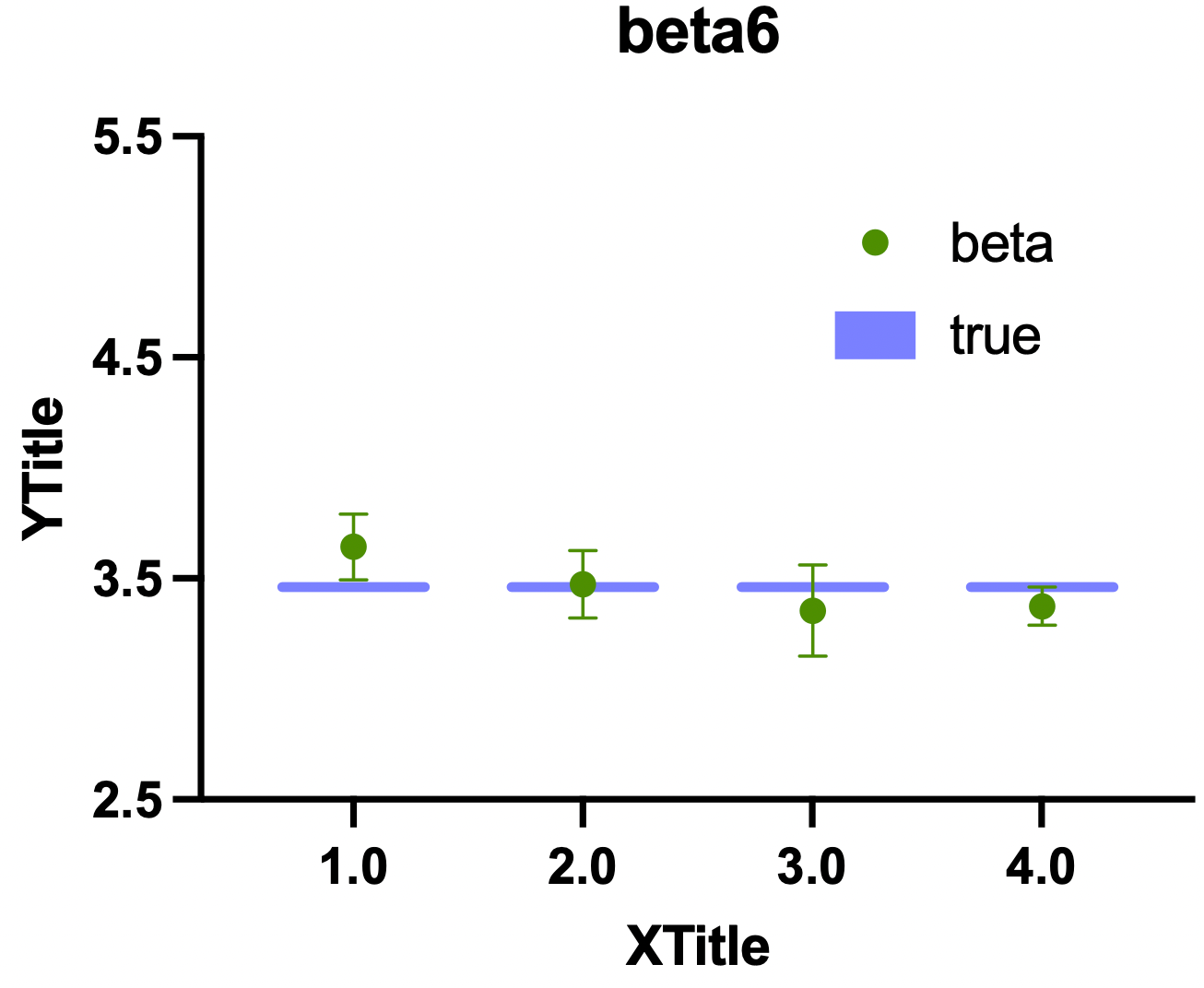
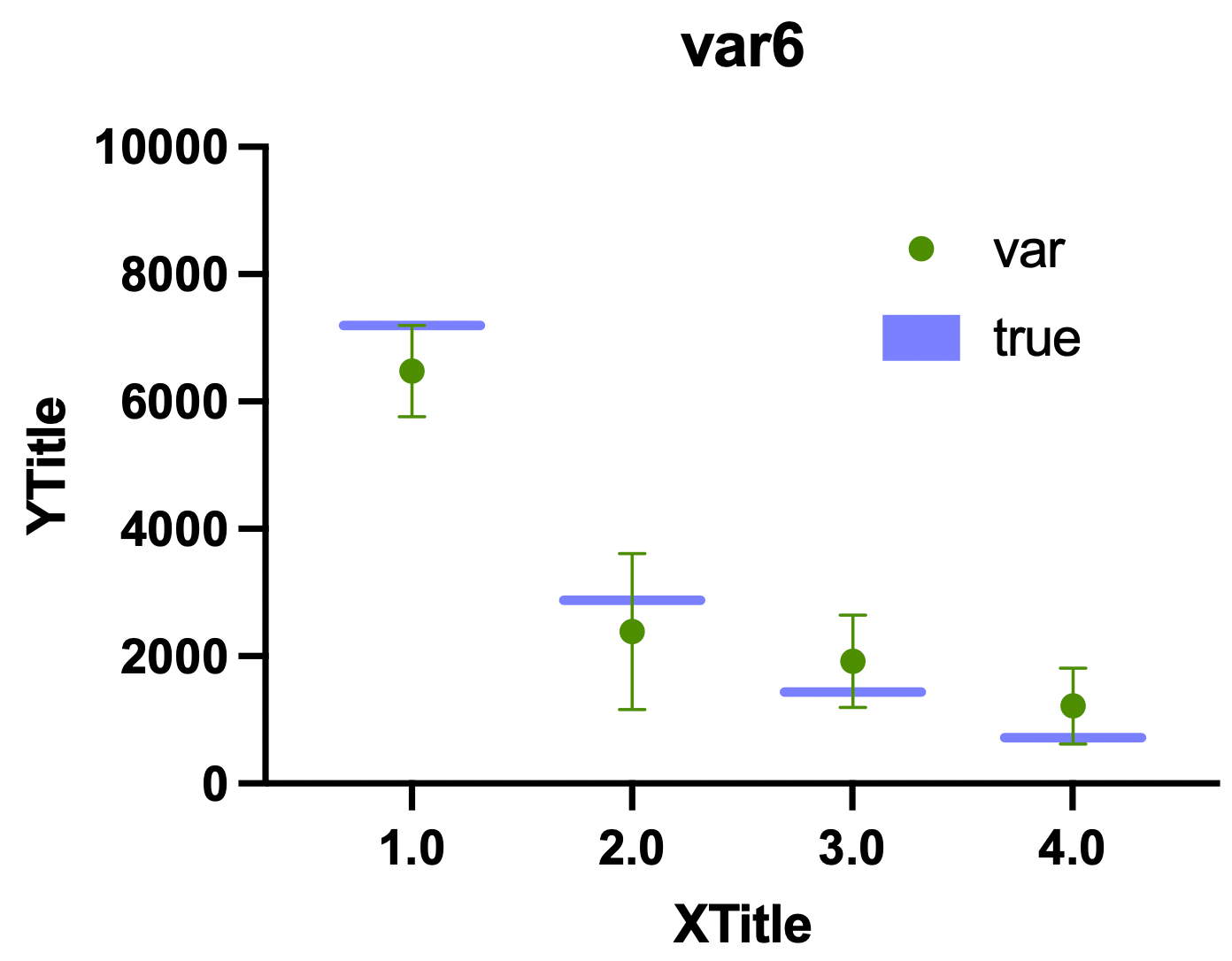
换做是5000次SSA，根本不行

究其原因，感觉是拟合效果不好导致，因此加入两组Attribute=0.52进行训练，拟合与预测效果会更好，Inference的结果也算可以接受

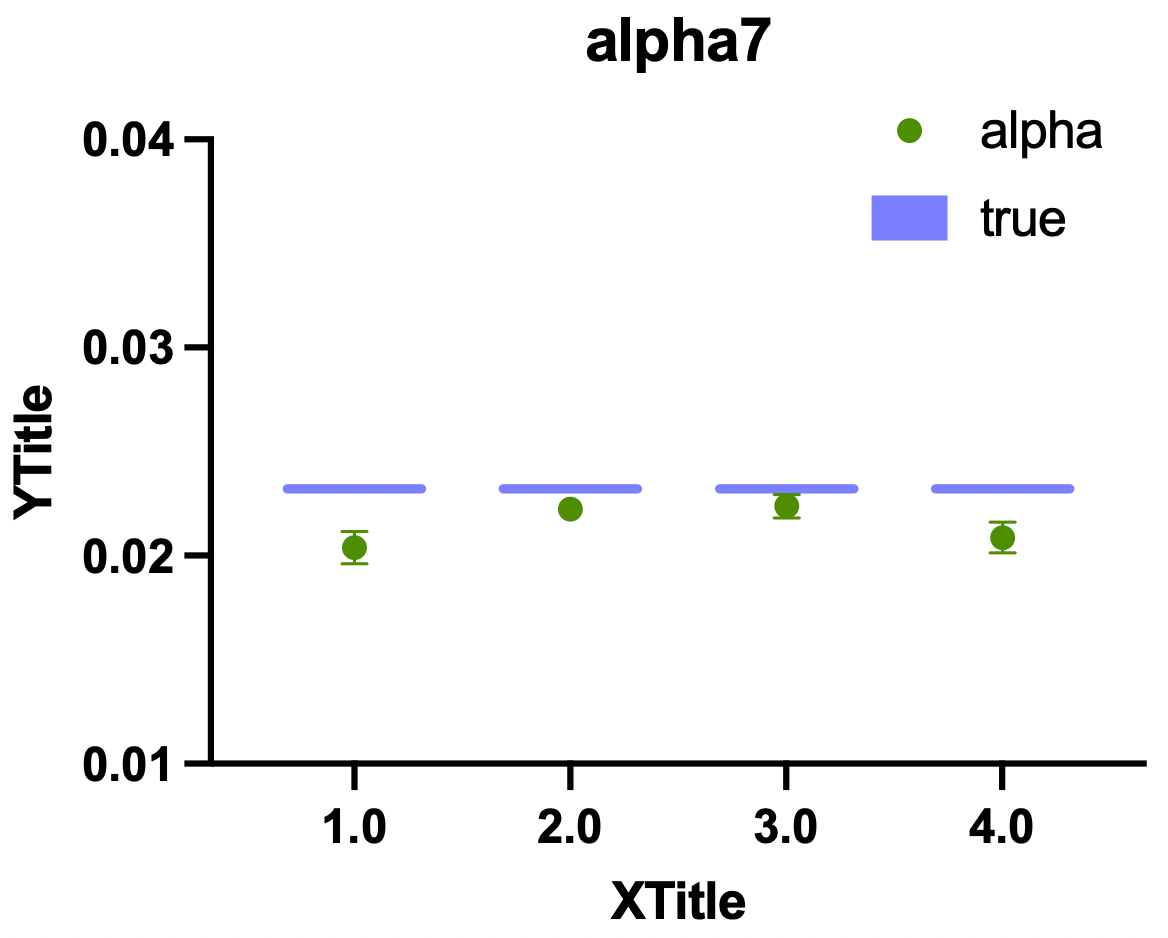
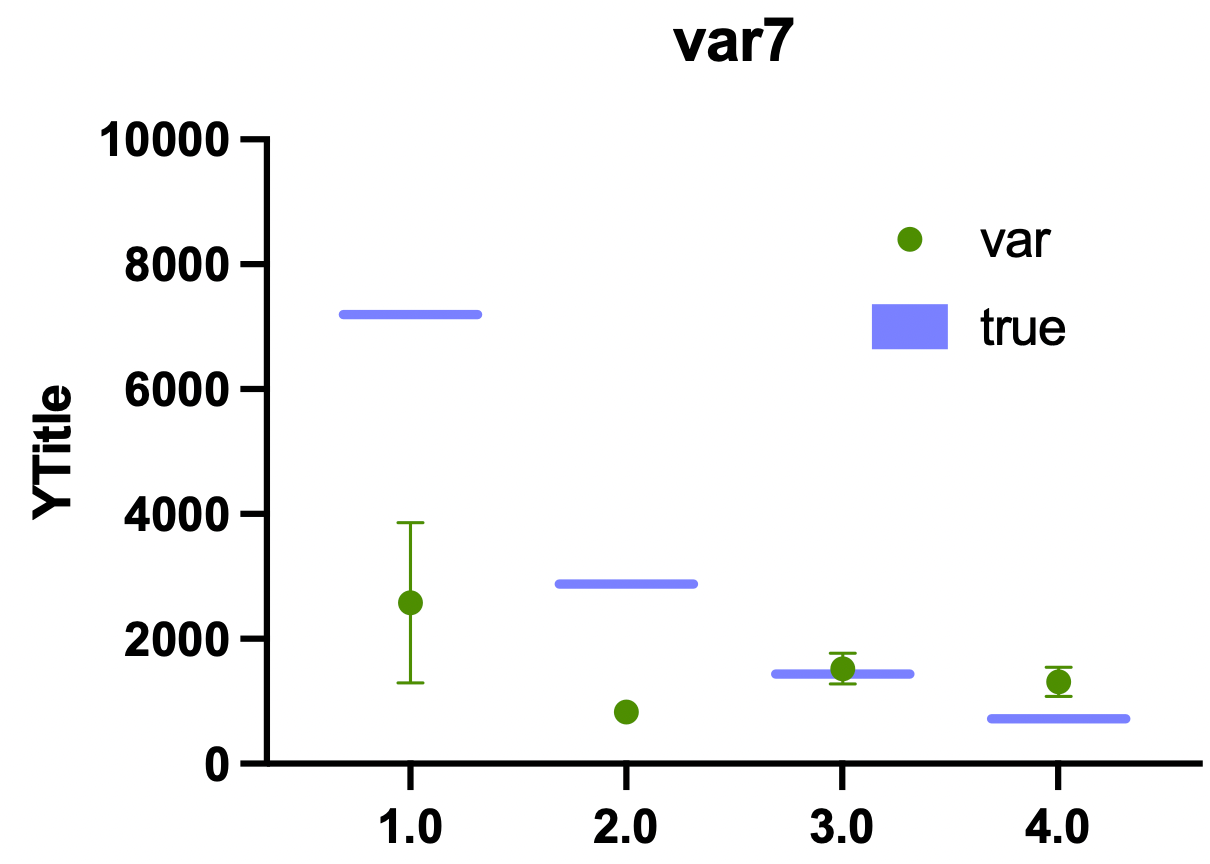
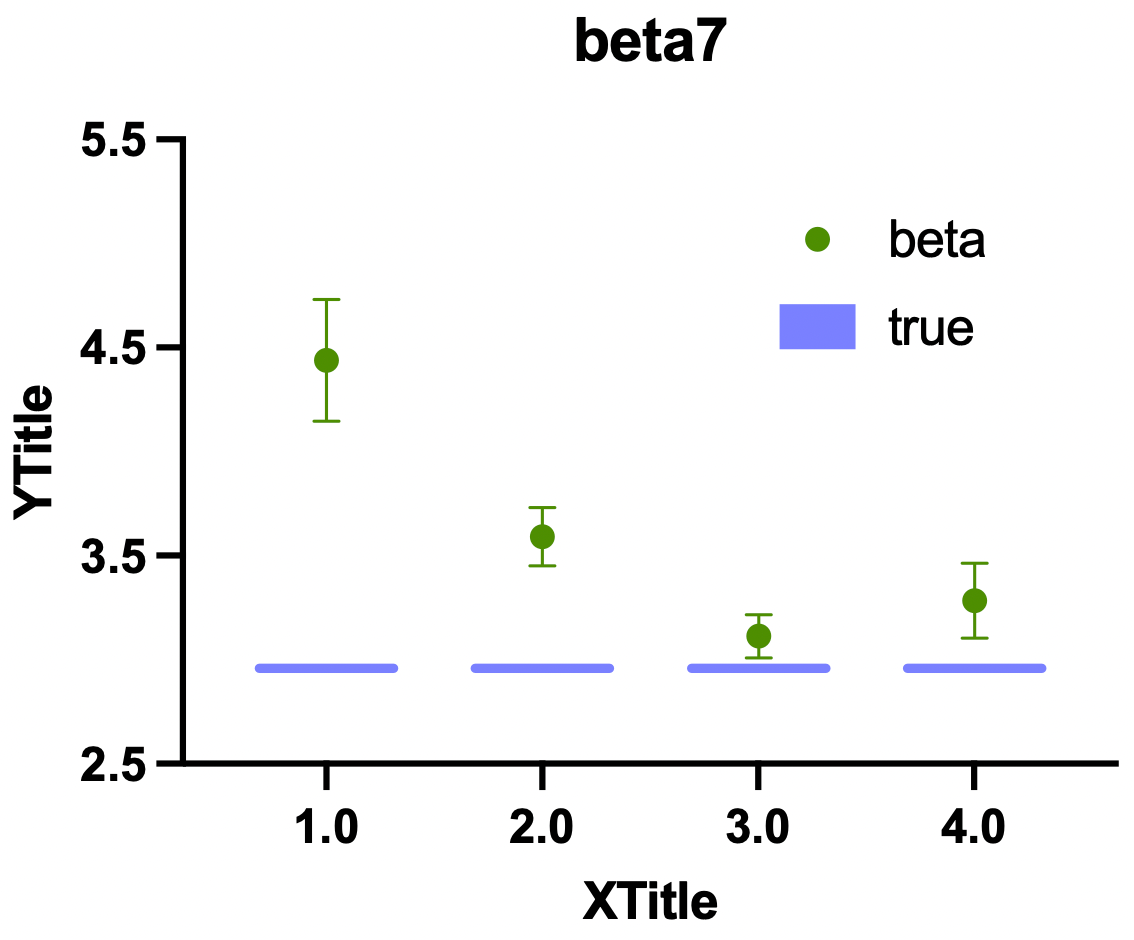




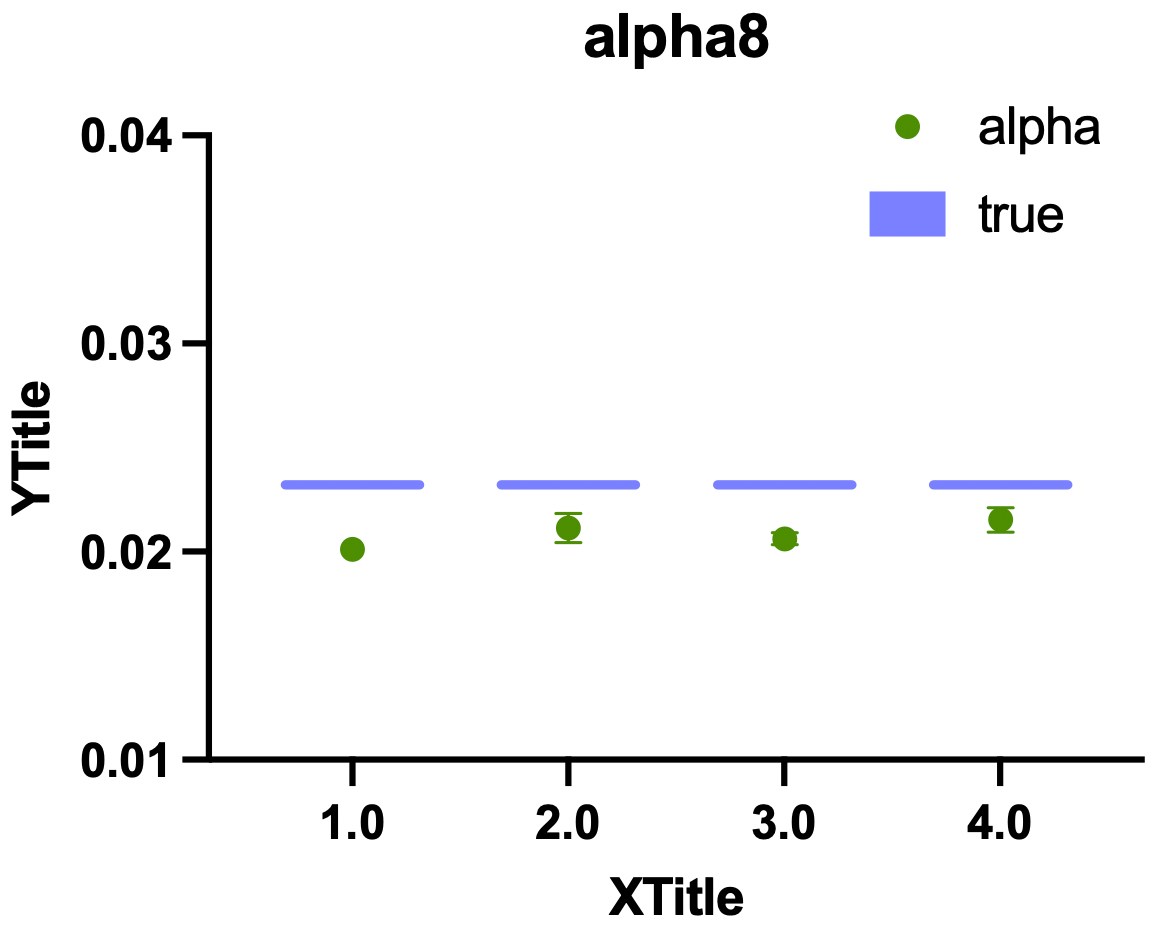
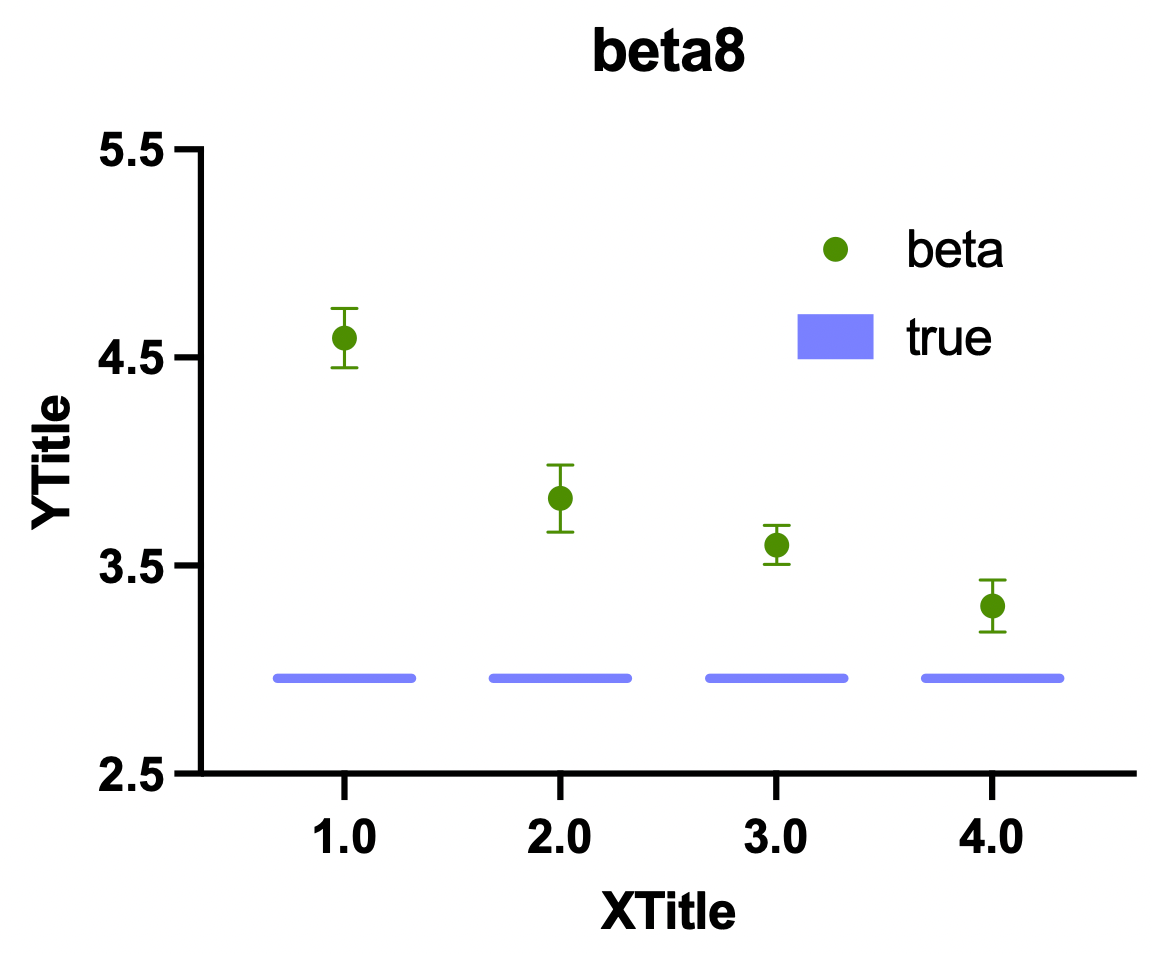
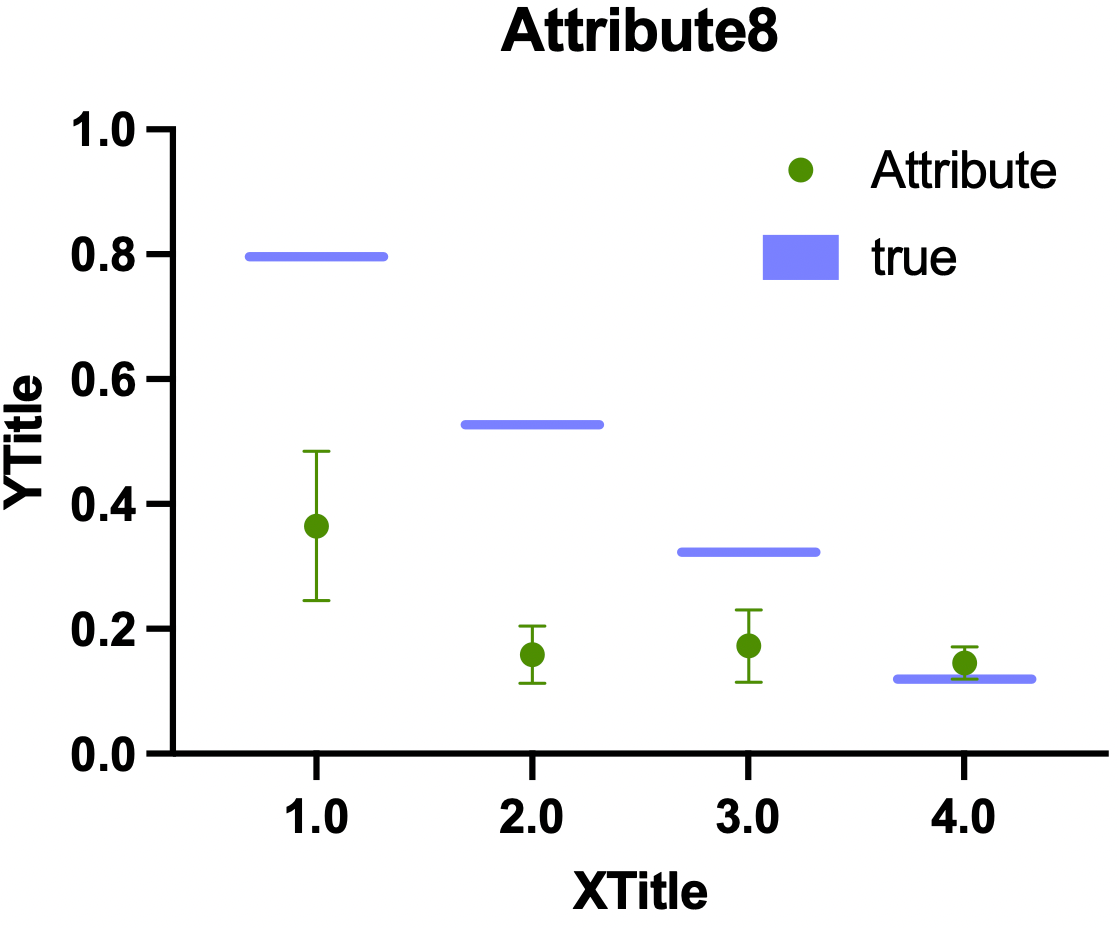
  

因此对alpha = 0.0232，beta = 2.96的概率分布进行Inference，用5k次SSA、50w次SSA，以及50w次SSA仅预测Attribute，结果均不可以接受

5k次SSA

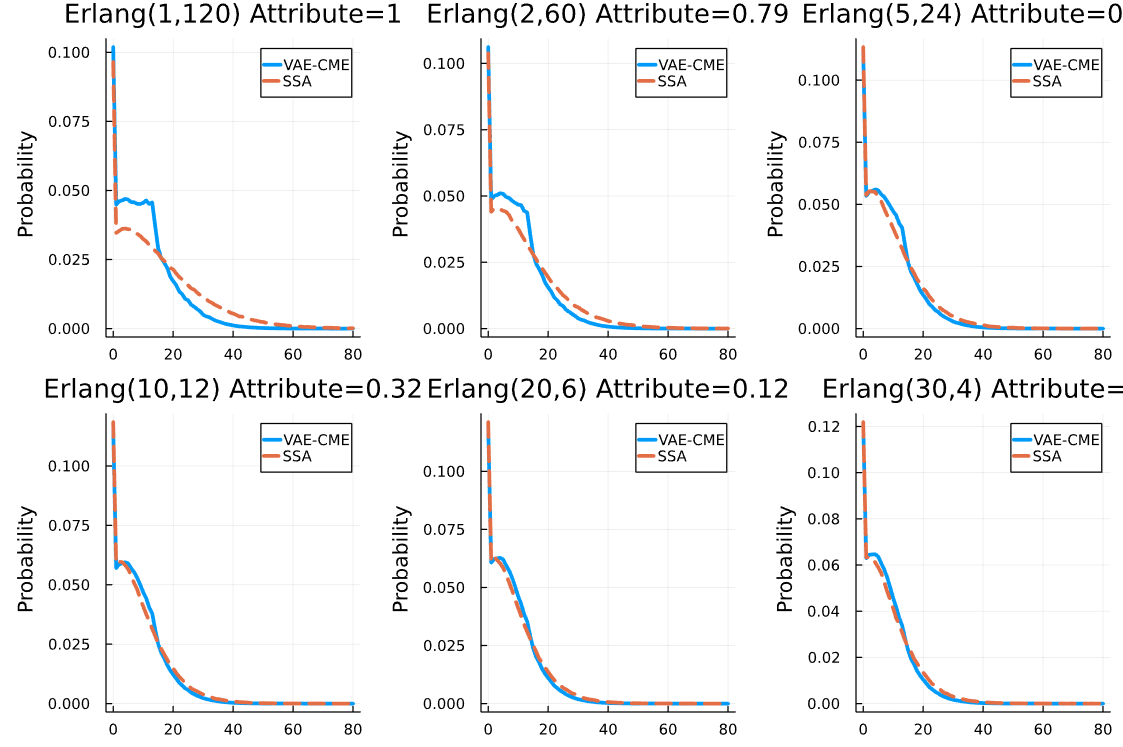
50w次SSA

50w次SSA，仅对Attribute的Inference效果



预测效果



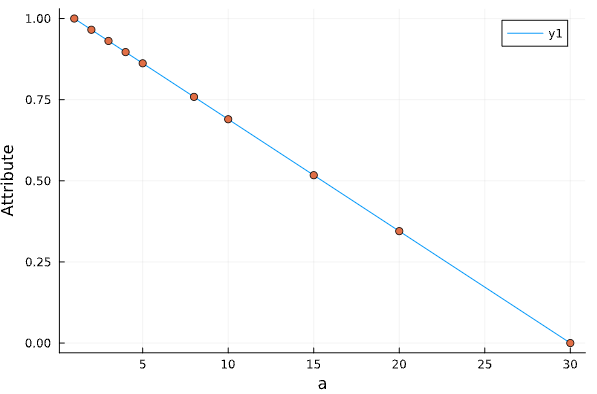
训练集

Erlang(1,120), Attribute = 1

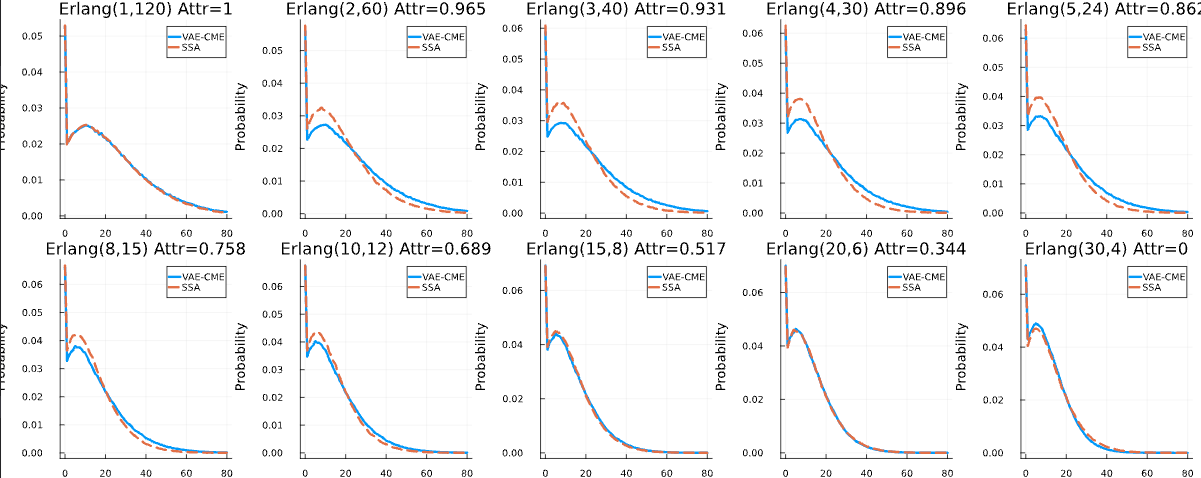
Erlang(15,8), Attribute = 0.5172

Erlang(30,4), Attribute = 0

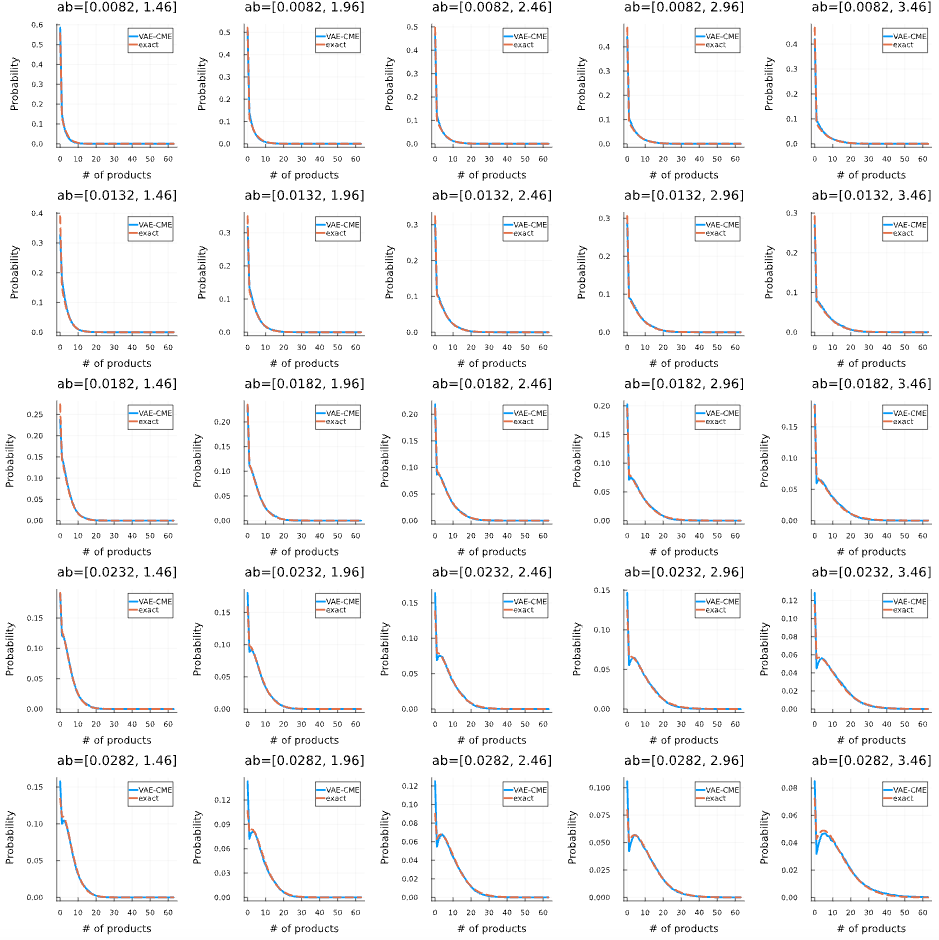
按照如下线性关系取



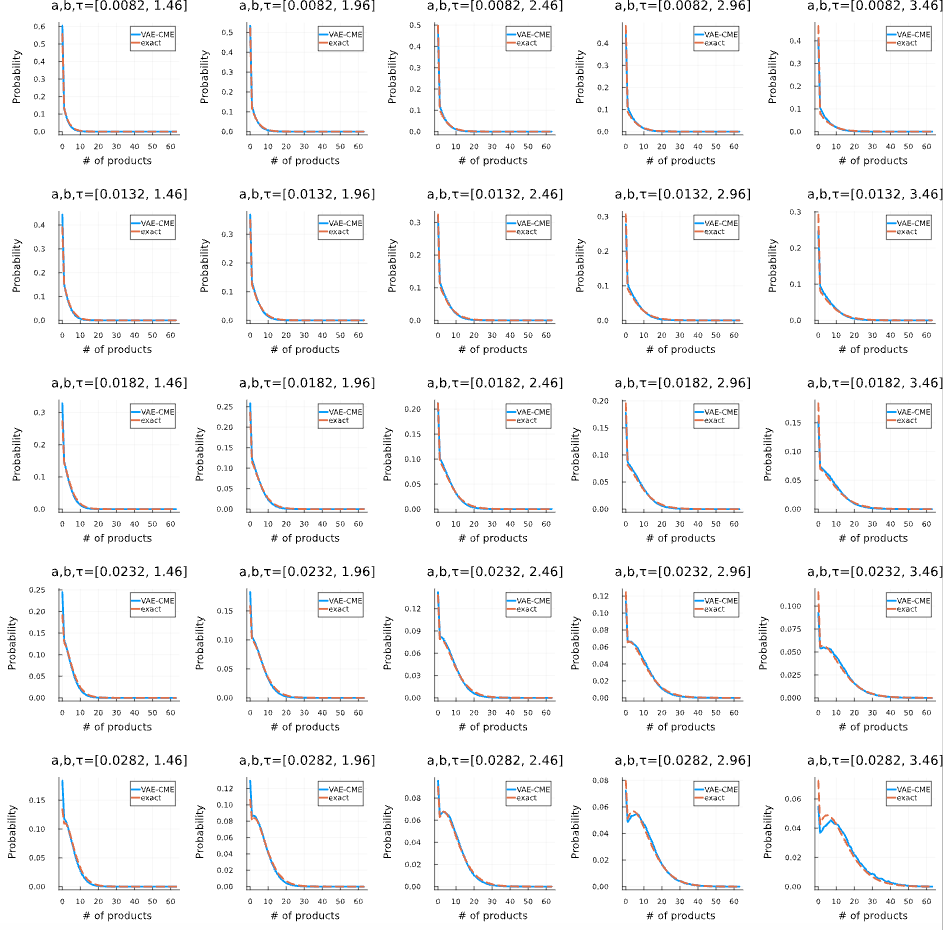
按照这样的线性关系，拟合和预测效果不行，不是起作用



增加alpha beta的情况，做了没有方差的情况，tau是固定值，训练集是四个角上的SSA



之前也做过25组alpha和beta的情况



很明显是达不到做Inference的拟合效果的

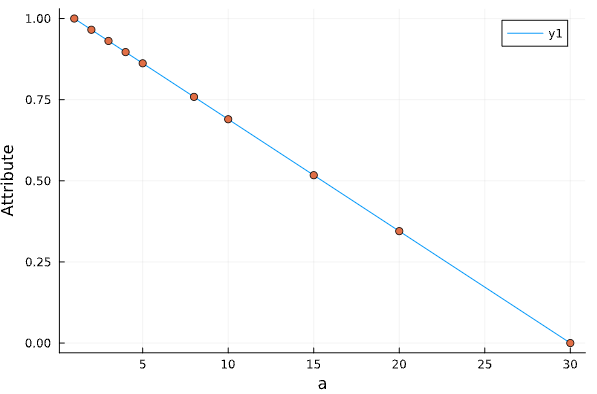
1、中间插一组训练集

Erlang(1,120), Attribute = 1

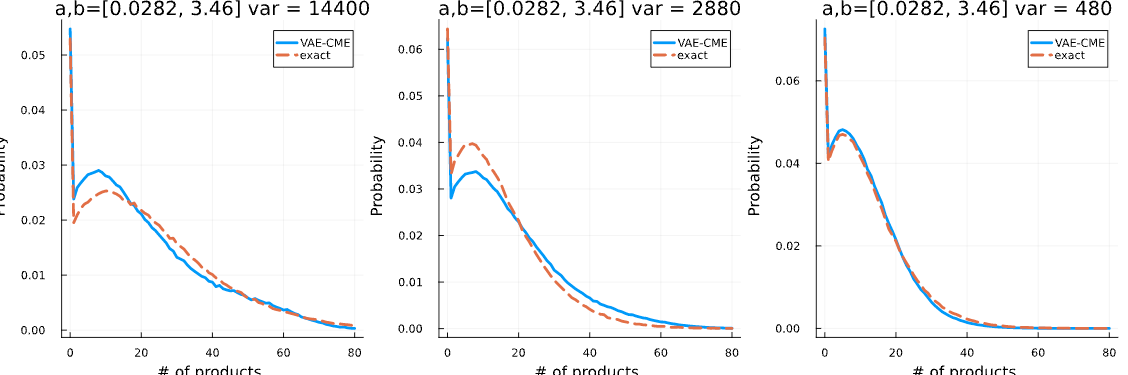
Erlang(5,24), Attribute = 0.8620

Erlang(30,4), Attribute = 0

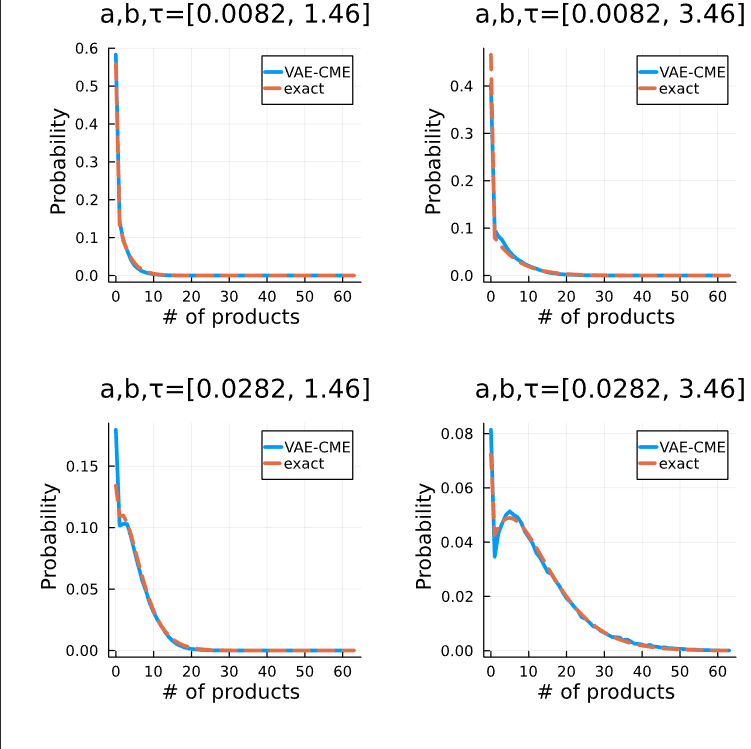
按照如下线性关系取



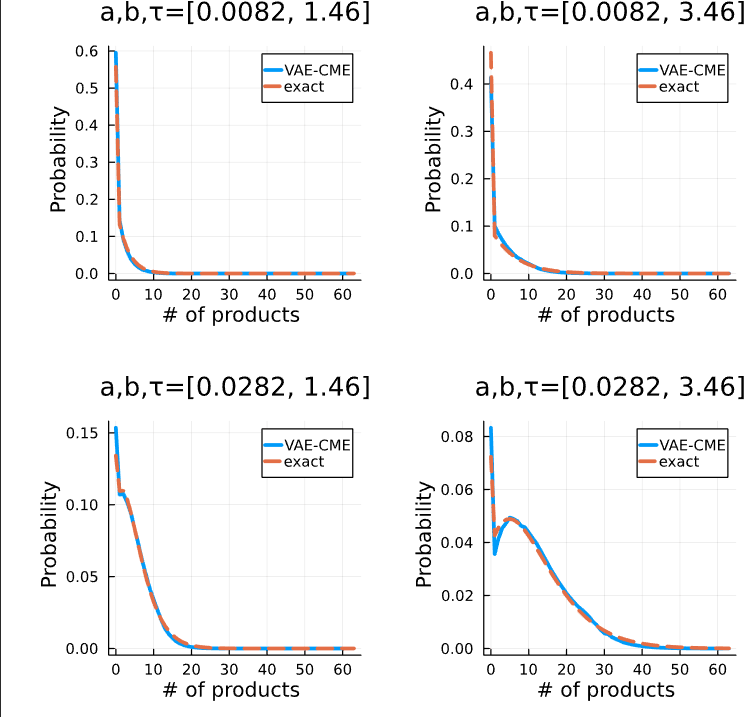
拟合不上



2、4组alpha和beta，增加0点的正则项

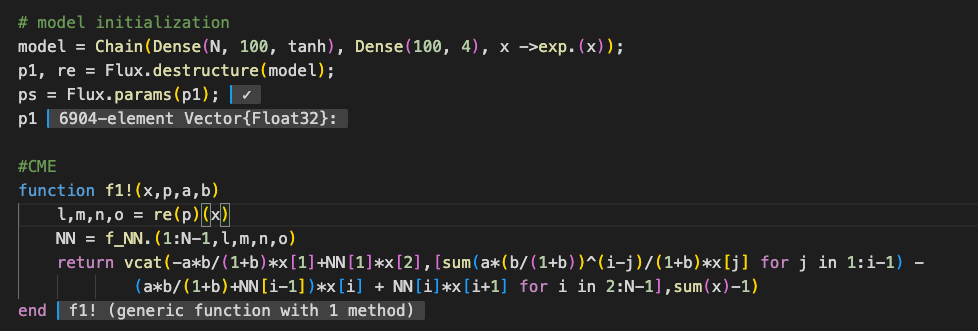


3、用bp训练4组alpha和beta，效果似乎和VAE是差不多的

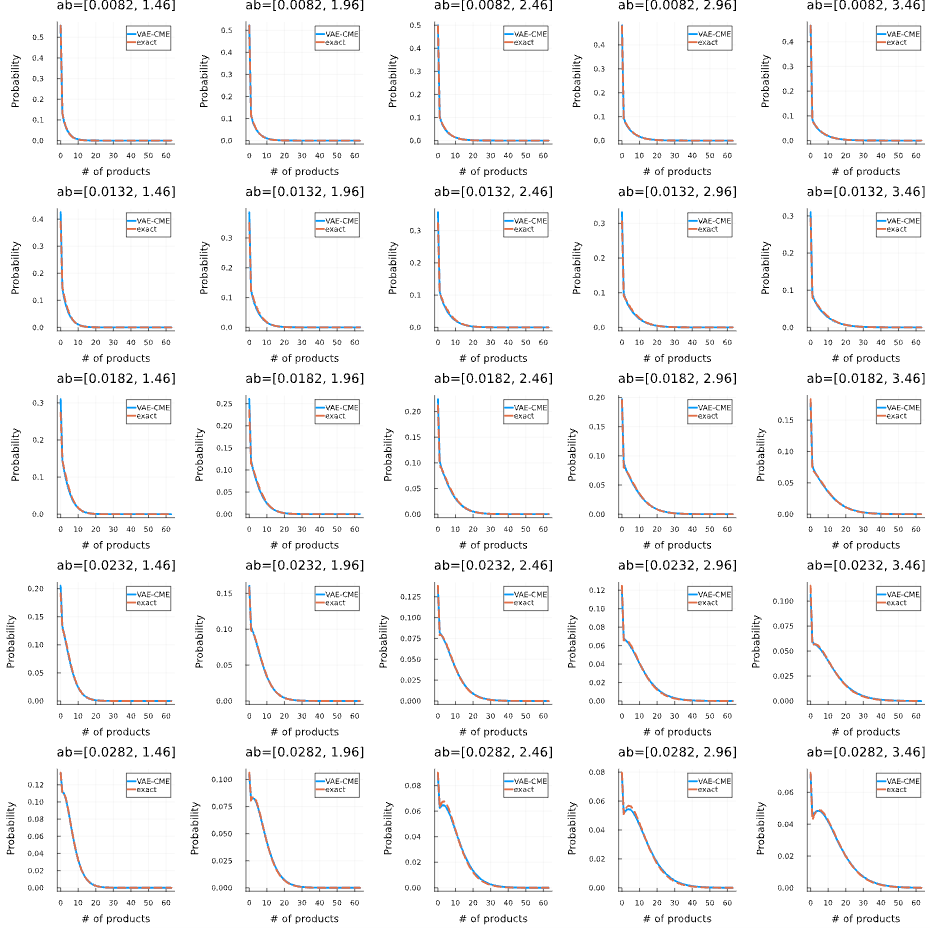


20231019

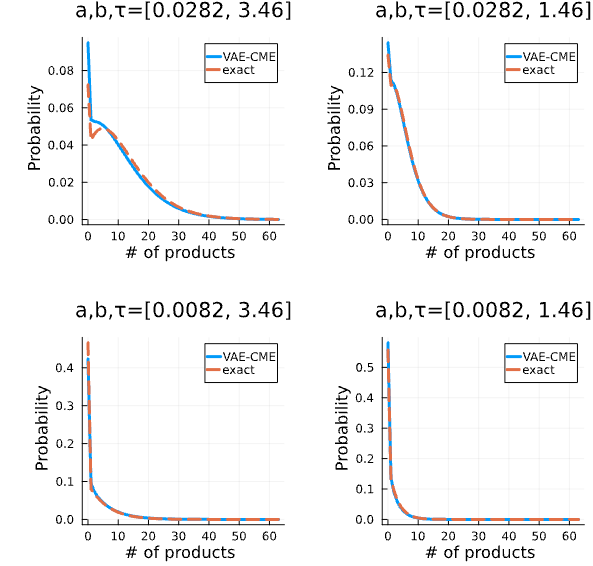
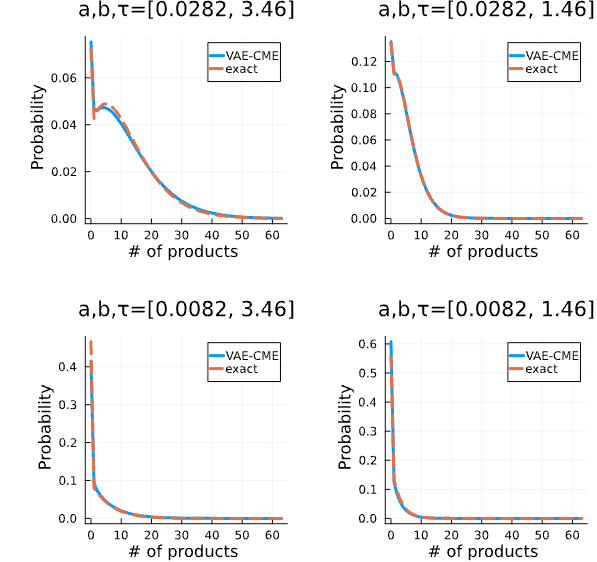
用MLP网络进行测试，改变网络结构增大隐藏层，输出层为四个参数，激活函数选择exp



输入四组alpha和beta，发现效果很好

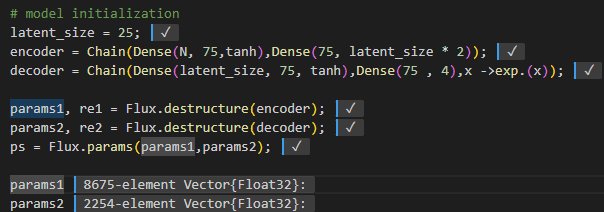


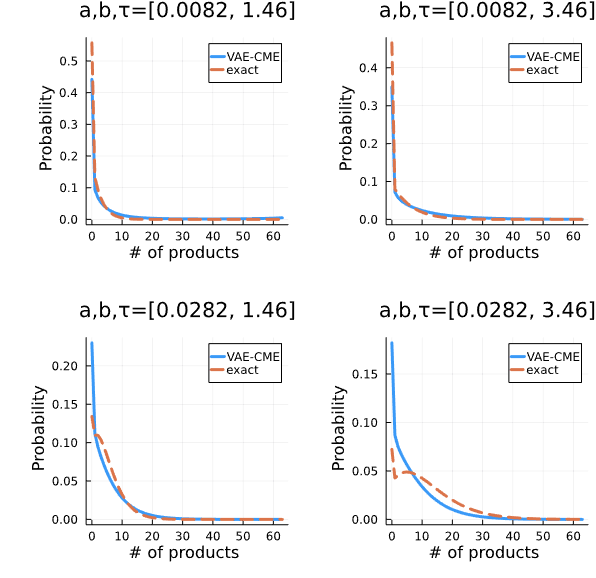
但这似乎并不是每次都能训到这么好的，重复做了两组实验，发现会有陷入局部最优的情况，时好时坏



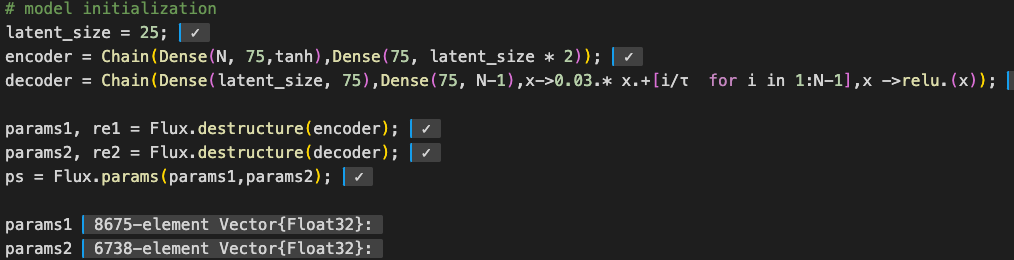
转到VAE，训练两个方法，

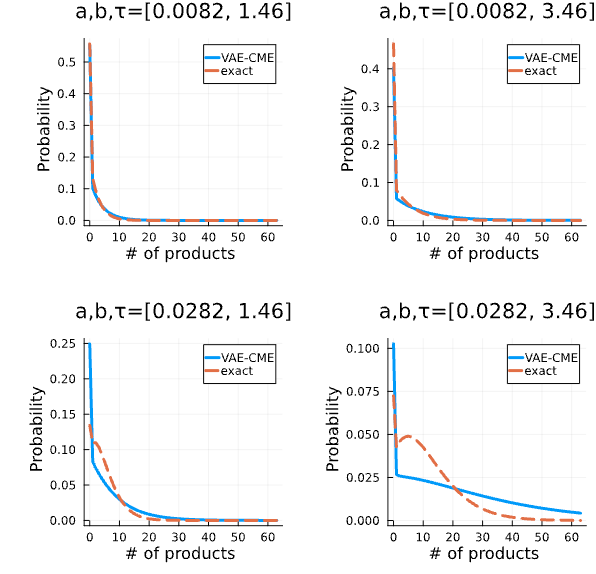
一是VAE的输出改为abcd，并增大隐藏层和z的维数，效果不行的



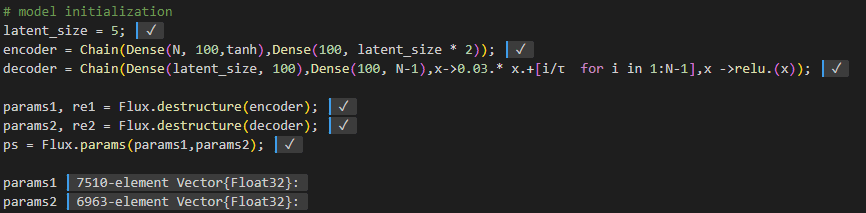


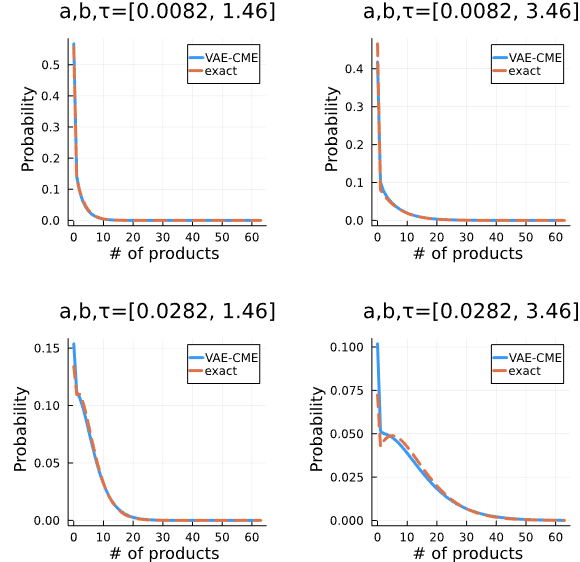
一仅仅增大隐藏层和z的维数，不用abcd，训练效果竟然变差了，不如之前较小的隐藏层和z



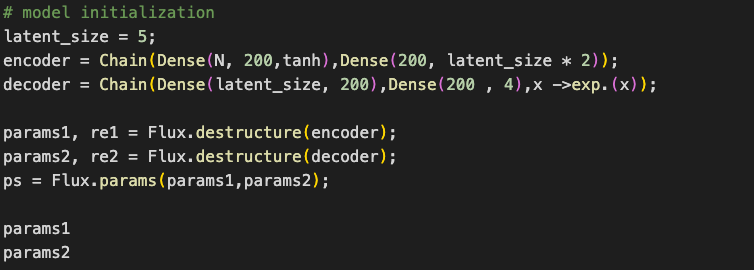


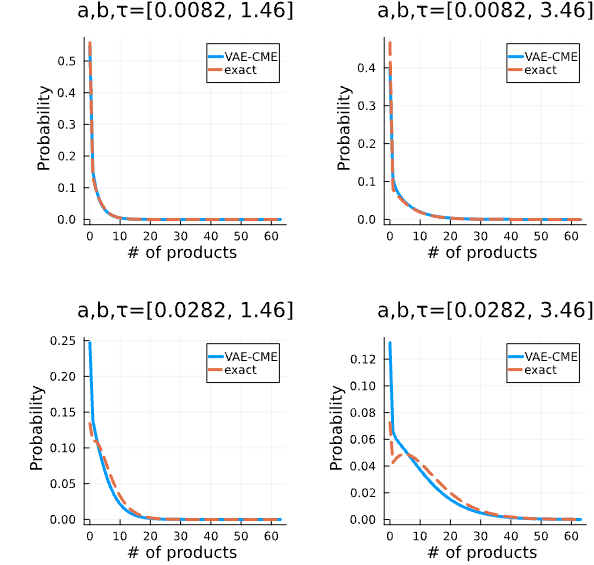
减小z的维数，效果与之前差不多，奇怪的是z维数小的情况下训练效果反而变好了，有点反常



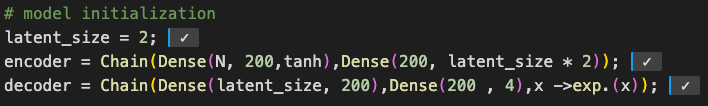


转到VAE的输出改为abcd，并只增大隐藏层维数，目前训练效果并不好

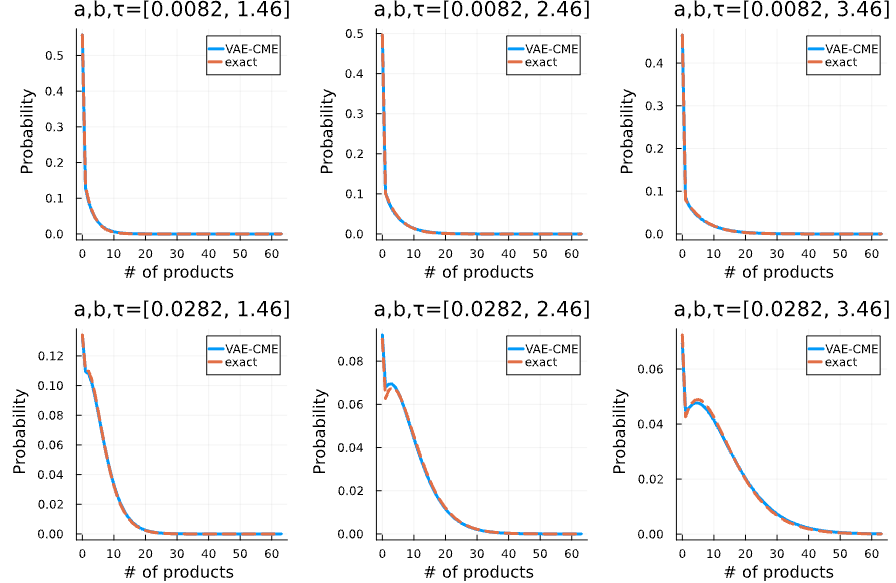




网络结构



输六组数据



可以预测

