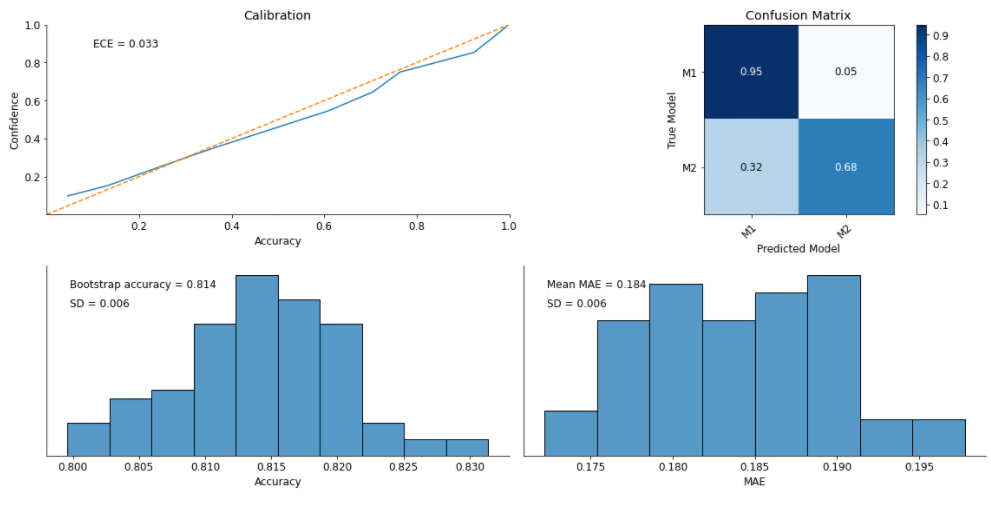
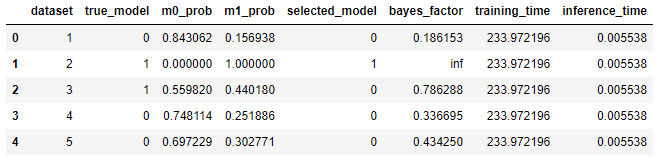
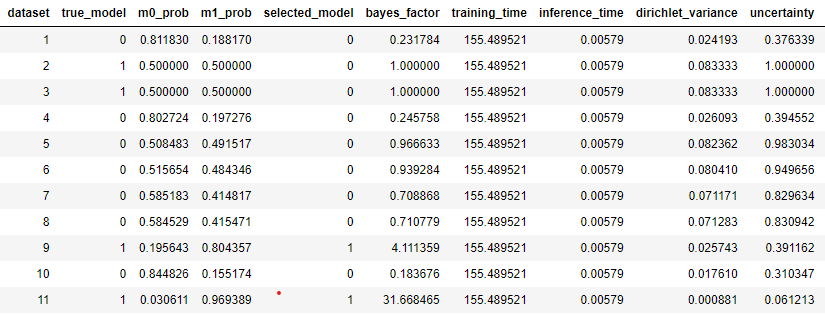
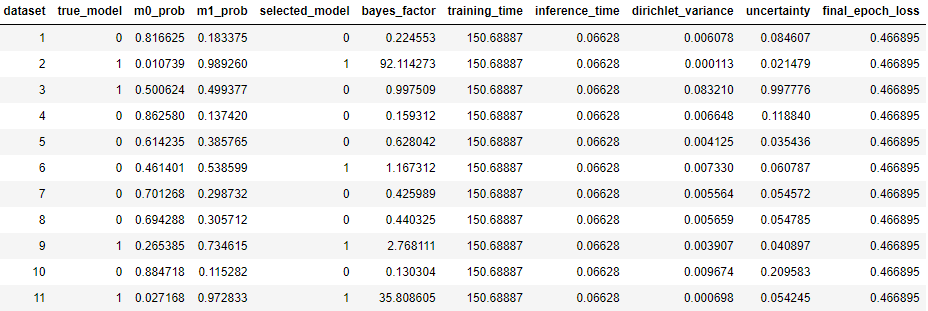
* + Performance-Abfall seit Implementierung der Dirichlet:
    - Vorher:
      * Accuracy: .81
      * ECE: .03
      * MAE: .18
      * 
      * 
    - 1. Run:
      * Accuracy auf .60
      * ECE & MAE verdoppelt
      * Network hat starken Bias zu Modell1 (true value = 0) 🡪 auch vorher schon da, jetzt extremer
      * 
      * Final epoch loss ~ .55
    - 2. Run:
      * Accuracy .75
      * ECE geringer! .02
      * MAE .24
      * 
    - 3. Run: vergleichbar zu erstem
    - Starting Loss entscheidend?
    - Probierte Lösungen
      * Mehr Iterations:
        + 1. Trial: Running Loss Unterschied von Epoch 10 (0.476) zu Epoch 20 (0.463) minimal 🡪 aber auch Epoch 1 schon bei .596 // Accuracy = .759
      * Learning rate: (vorher: 0.0005)
        + 0.005:

1. Run bei .69 gestartet, bei .54 fertig 🡪 Accuracy = .69

2. Run bei .545 gestartet, bei .539 fertig 🡪 Accuracy = .73

3. Run bei .562 gestartet, bei .466 fertig 🡪 Accuracy = .763

4. Run: .651 🡪 .466 // Acc = .761

5. Run: . 🡪 . // Acc = .

* + - * + 0.0001:

1. Run bei .650 gestartet, bei .545 fertig 🡪 Accuracy = .616

2. Run: .69 🡪 .55 // Acc = .638

* + - * + 0.001:

1. Run: .607 🡪 .545 // Acc = .70

2. Run: .606 🡪 .470 // Acc = .77

* + - * + 0.0007:

1. Run: .62 🡪 .54 // Acc = .64

2. Run: .54 🡪 .537 // Acc = .63

* + - * # Neuronen
        + 'dense\_inv\_pre\_pooling\_args' (vorher: 4 / BayesFlow: 32)

1. Run: .525 🡪 .471 // Acc = .767

2. Run: .578 🡪 .469 // Acc = .774

3. Run: . 🡪 . // Acc = .

* + Dirichlet-Distribution:
    - Wir vergleichen bei Experimenten ja immer nur 2 Modelle 🡪 da modellieren wir dann eigentlich Beta-Distribution?
    - Sollten individual concentration parameters immer noch lower bounded by 1 lassen oder erlauben wir auch kleiner 1?
    - Bayesflow: sample() Funktion in EvidentialNetwork
      * Wozu wird das samplen der model probabilities genutzt (experience replay?) und wir brauchen sie nicht, right?
    - HierarchicalEvidentialNetwork: predict() nutzt alpha = self.call(x)
      * Wie besser machen?