Likelihood fits with variable resolution in time-dependent charm measurements at LHCb

Francesco Terzuoli

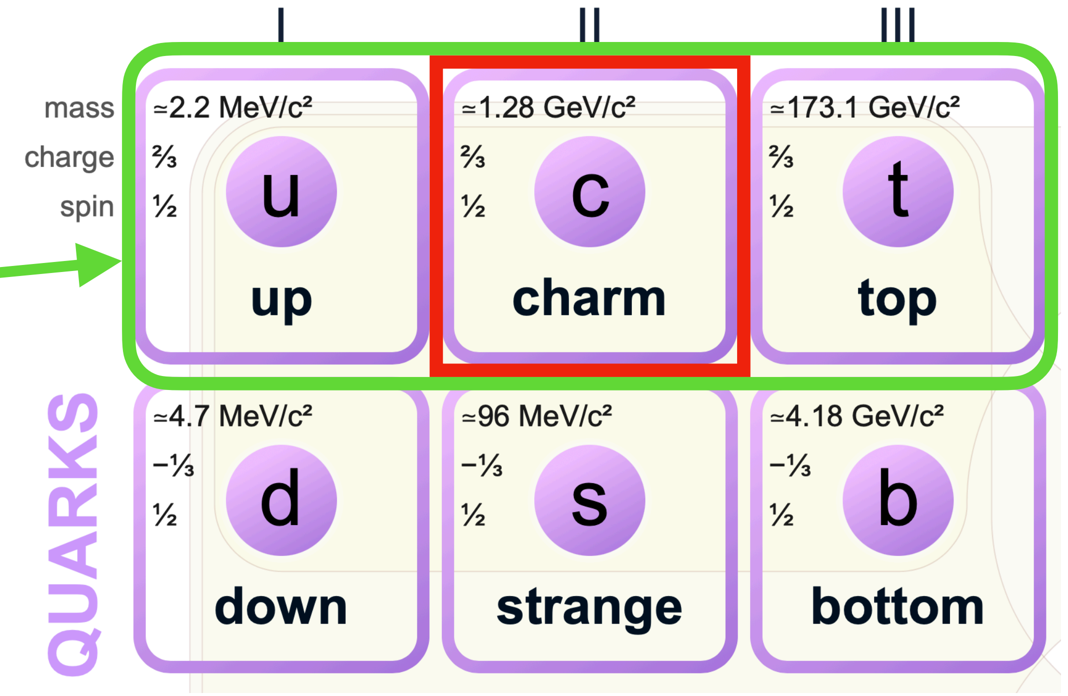
September Statistic Seminar

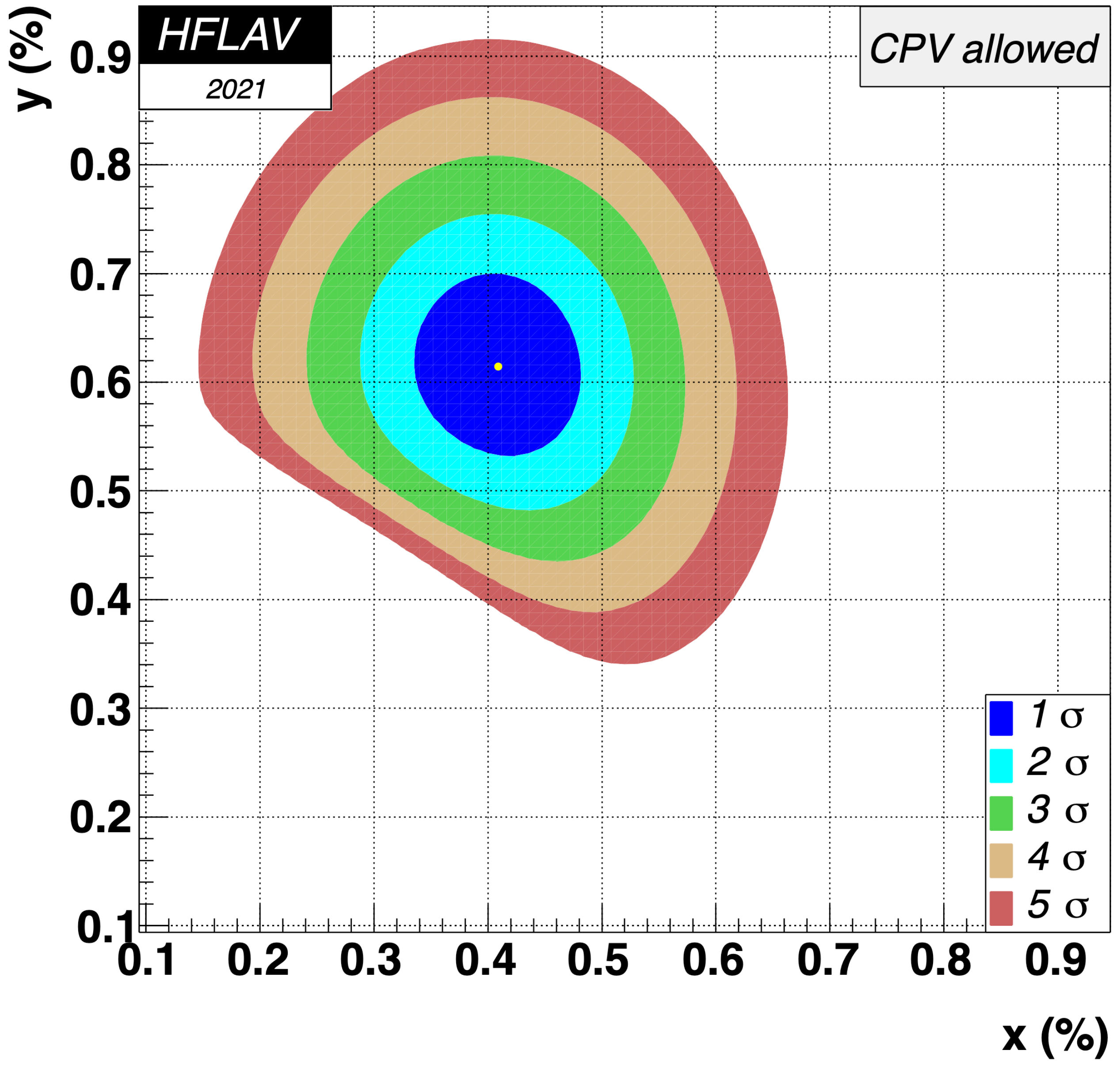
28/09/2023

Charm mixing and CP violation

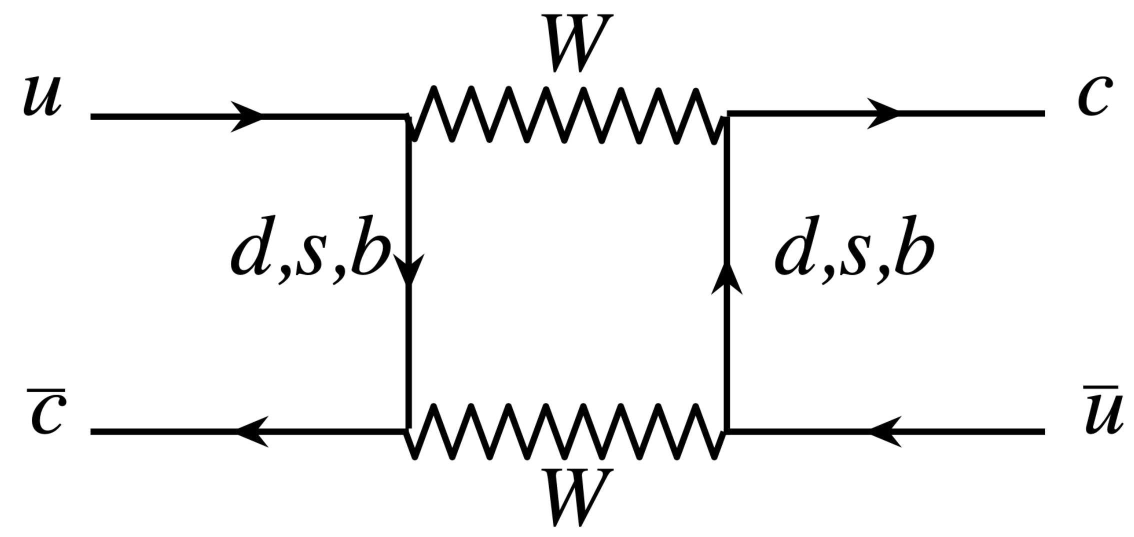
Why the charm sector is important?

- Interesting for CP violation measurements
  - Spacial inversion (P) + charge conjugation (C) → CP
- Unique position among up-quarks:
  - t quark hadronizes before decaying
  - u quark is too light for interesting processes

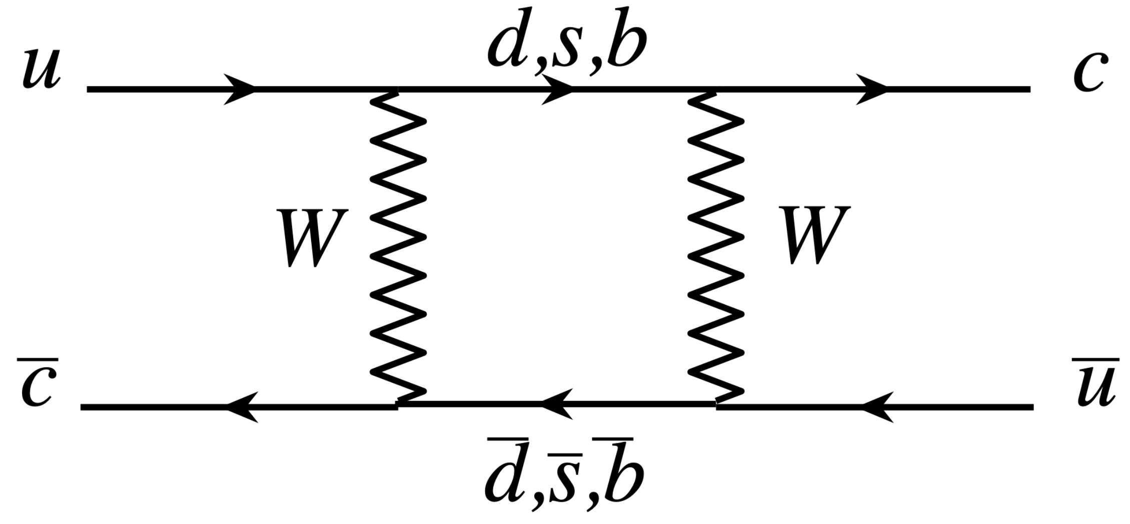




Short distance contribution (off-shell) → x



Long distance contribution (on-shell) → y

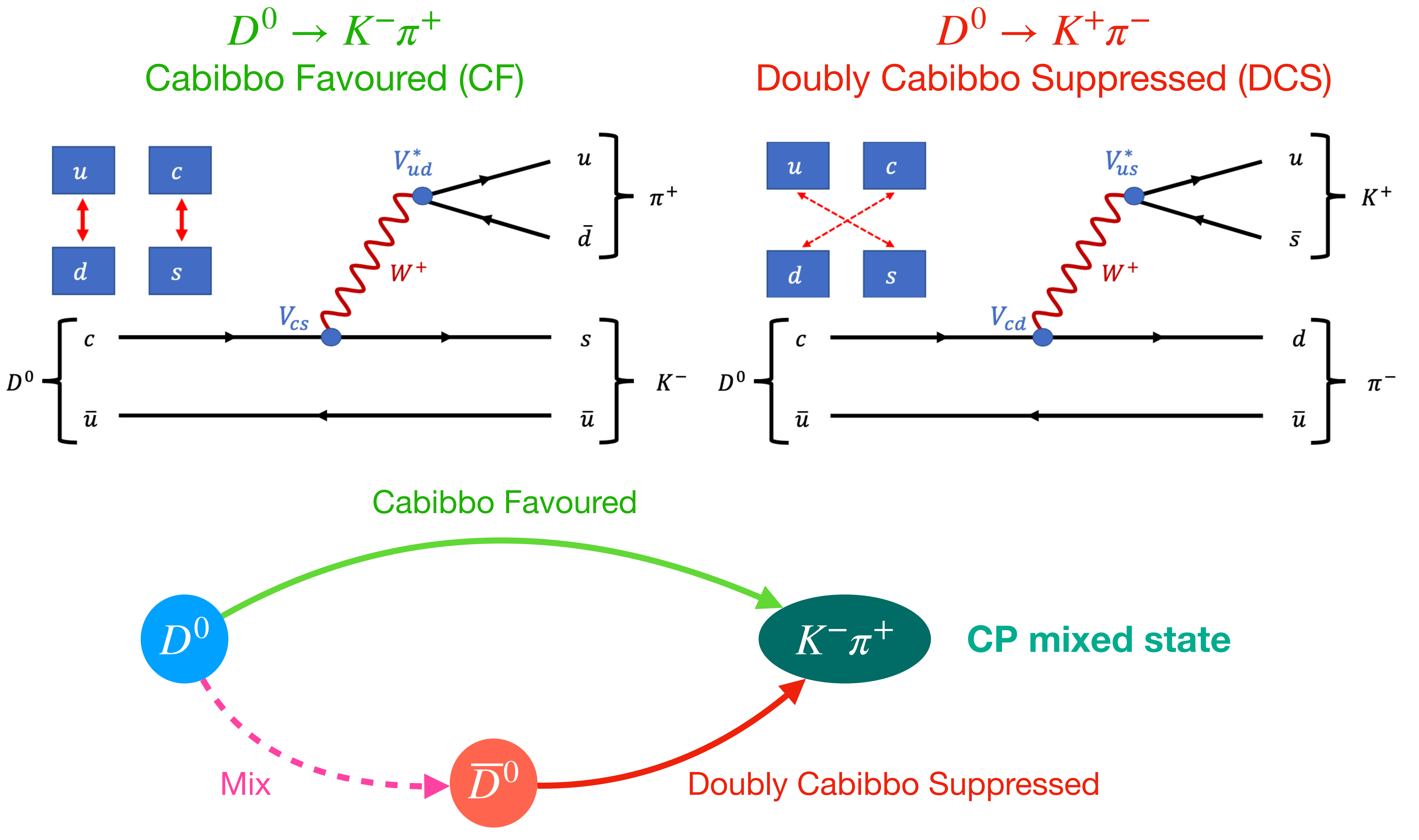


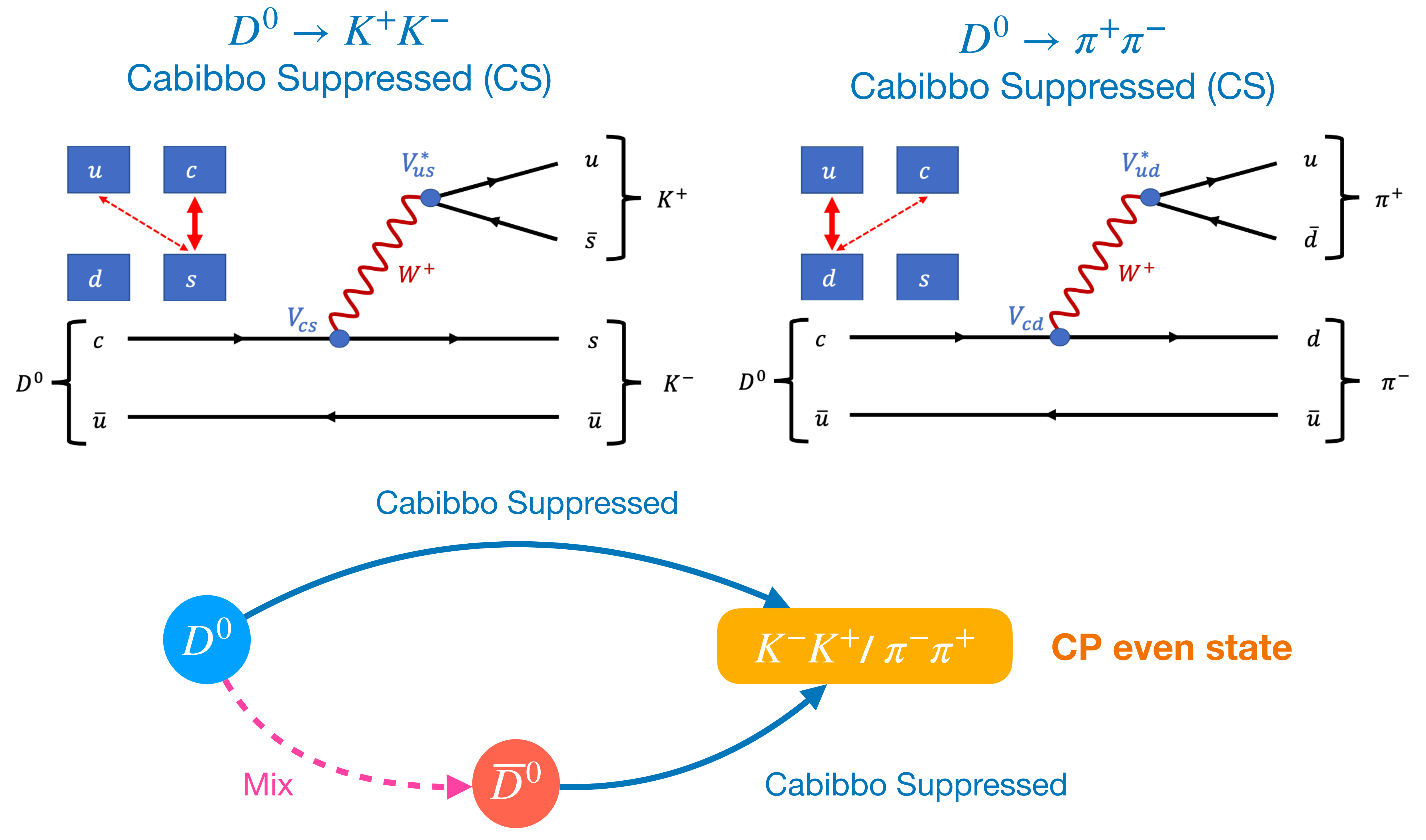
Mixing in the charm sector

- D^0 mixing due to mass eigenstates $\neq$ flavour eigenstates: $|D_{1,2}\rangle = p|D^0\rangle + q|\overline{D}^0\rangle$
- Oscillation (mixing) parameters: $x = \dfrac{m_1 - m_2}{\Gamma}$ and $y = \dfrac{\Gamma_1 - \Gamma_2}{2\Gamma}$ with $\Gamma = \dfrac{\Gamma_1 + \Gamma_2}{2}$
- Oscillation probability $\mathscr{P}\left(D^0 \to \overline{D}^0(t)\right) = \left|q/p\right|^2 \dfrac{e^{-\Gamma t}}{2}\left(\cosh(y\Gamma t) - \cos(x\Gamma t)\right)$

Charm mixing and CP violation

Mixing in D^0 two-body decay





- Interference between $D^0 - \overline{D}^0$ mixing and decay leads to different time-dependent decay rates for each final state:
  - Different measured lifetimes between $D^0 \to K^-\pi^+$ and $D^0 \to h^-h^+$, with $h = K, \pi$
  - Probe mixing dynamics through $\tau\left(D^0 \to K^-\pi^+\right)$ and $\tau\left(D^0 \to h^-h^+\right)$

Charm mixing and CP violation

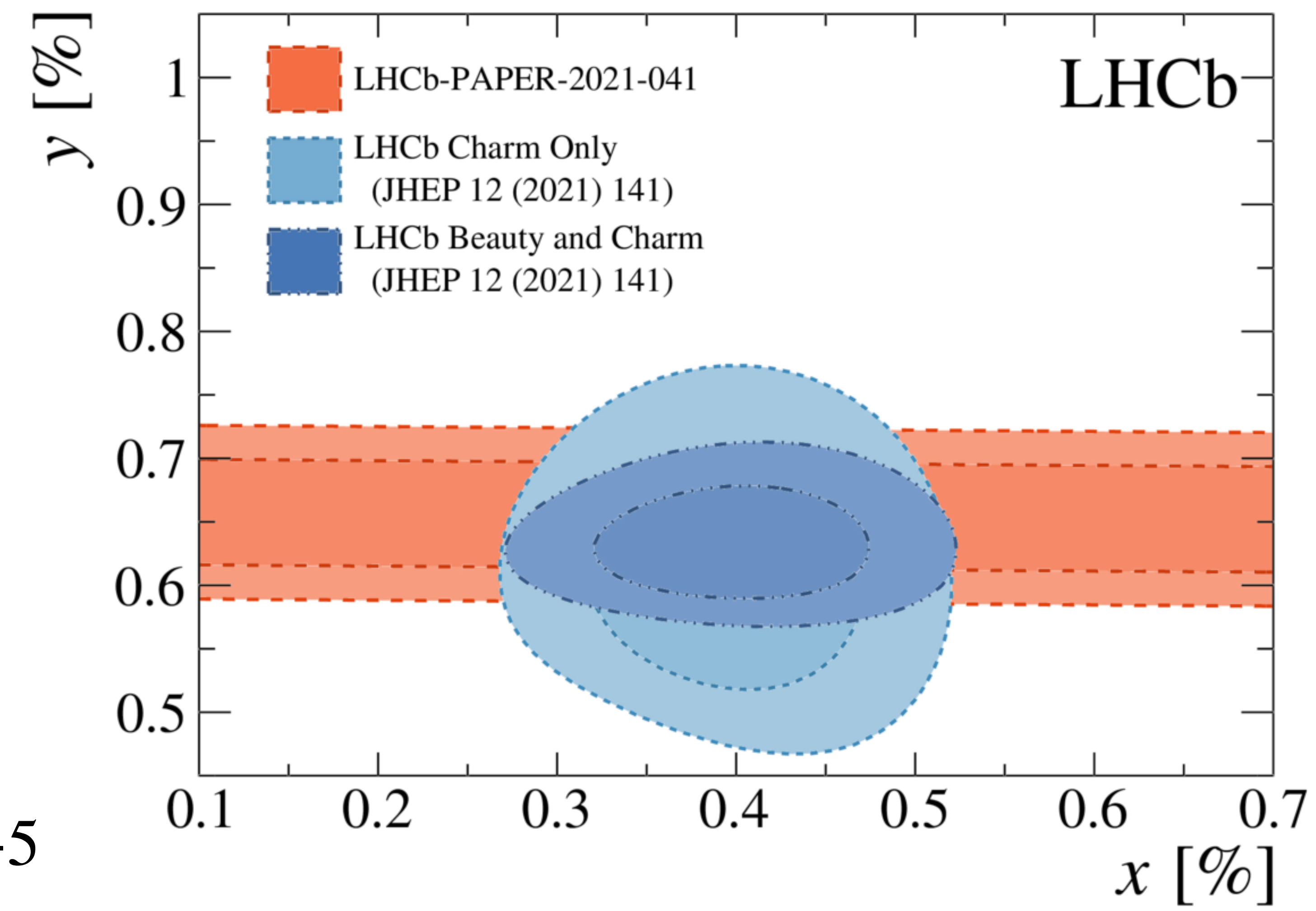
CP violation in D^0 two-body decay: the y_{CP} observable

- Given a CP even final eigenstate f, it can be defined:

$$y_{CP} \equiv \frac{\hat{\Gamma}\left(D^0 \to f\right) + \hat{\Gamma}\left(\overline{D}^0 \to f\right)}{2\Gamma} - 1 \approx \frac{\tau(D^0 \to K^-\pi^+)}{\tau(D^0 \to f)} - 1,$$

with $\hat{\Gamma}$ effective decay width from $\Gamma\left(D^0 \to f, t\right) \approx e^{-\hat{\Gamma}(D^0 \to f)t}$

- From SM, CP conservation $|y_{CP} - y| = 0$ and CP violation $|y_{CP} - y| \lesssim 3 \times 10^{-5}$



- y_{CP} both probe for additional sign of CPV in charm sector and also for New Physics

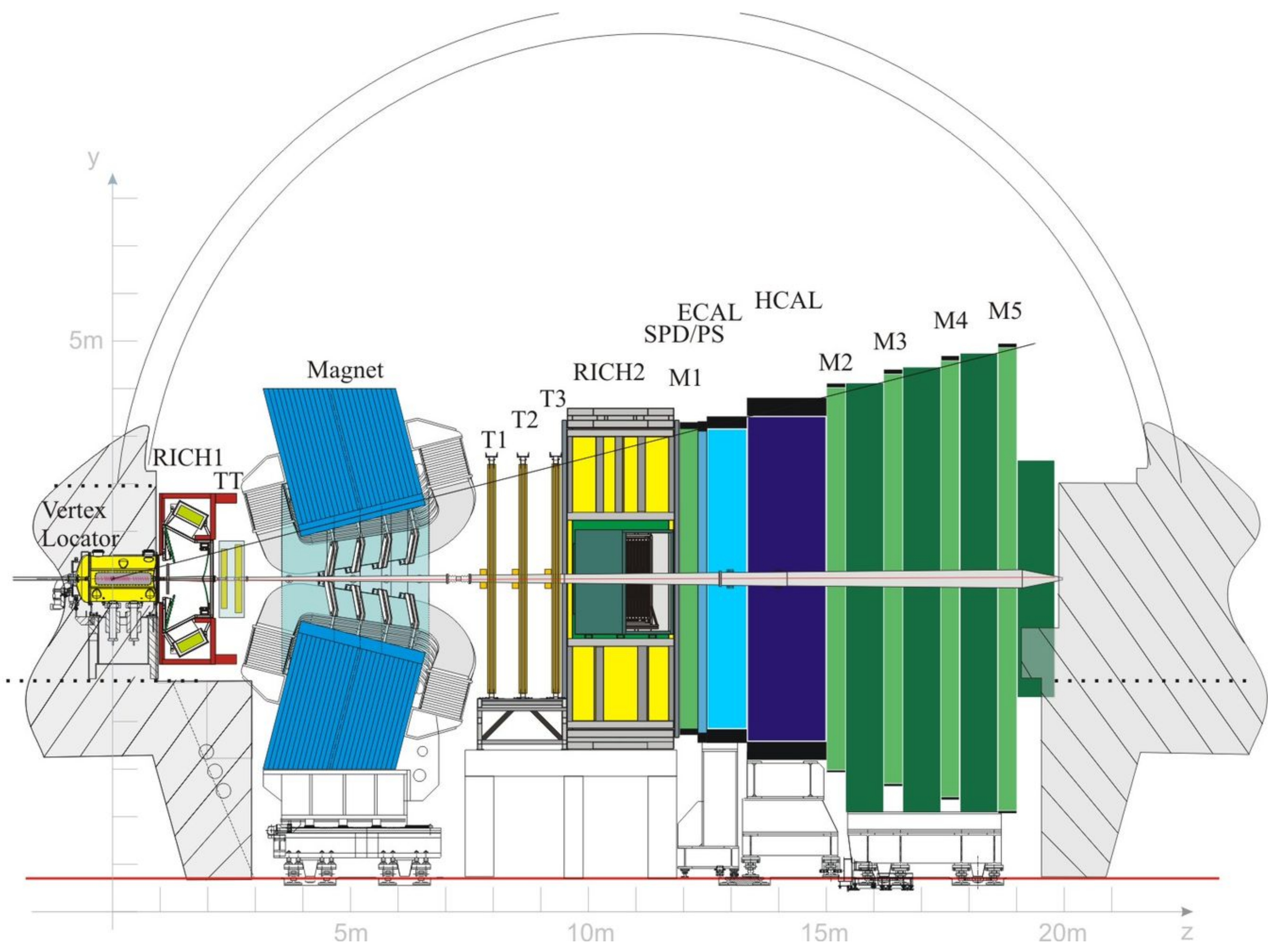
$y = (6.15^{+0.56}_{-0.55}) \times 10^{-3}$
from HFLAV

- Current best y_{CP} measurement: LHCb 25^{th} May 2022 $y_{CP} = (6.96 \pm 0.26_{stat} \pm 0.13_{syst}) \times 10^{-3}$

- Much upcoming data and measurement not improvable due to lack of knowledge of detector response

The LHCb detector

The detector characteristics



- Designed for beauty and charm physics at LHC (pp/ions collider @ CERN)
  - Single-arm forward spectrometer
  - Efficient geometry heavy flavour physics $1.8 < \eta < 4.9$
- Tracking system
  - VErtex LOcator (VELO): silicon strip vertex detector
  - Depending on where the particle decays the detector response varies → bias on decay time distribution
- Particle identification (RICH+calorimeters)
  - Different responses between K and π
- Trigger system → from $30MHz$ to $12.5kHz$
  - High charm production rate and limited bandwidth

1. Very tight selection
2. Pre-scaled samples with much background

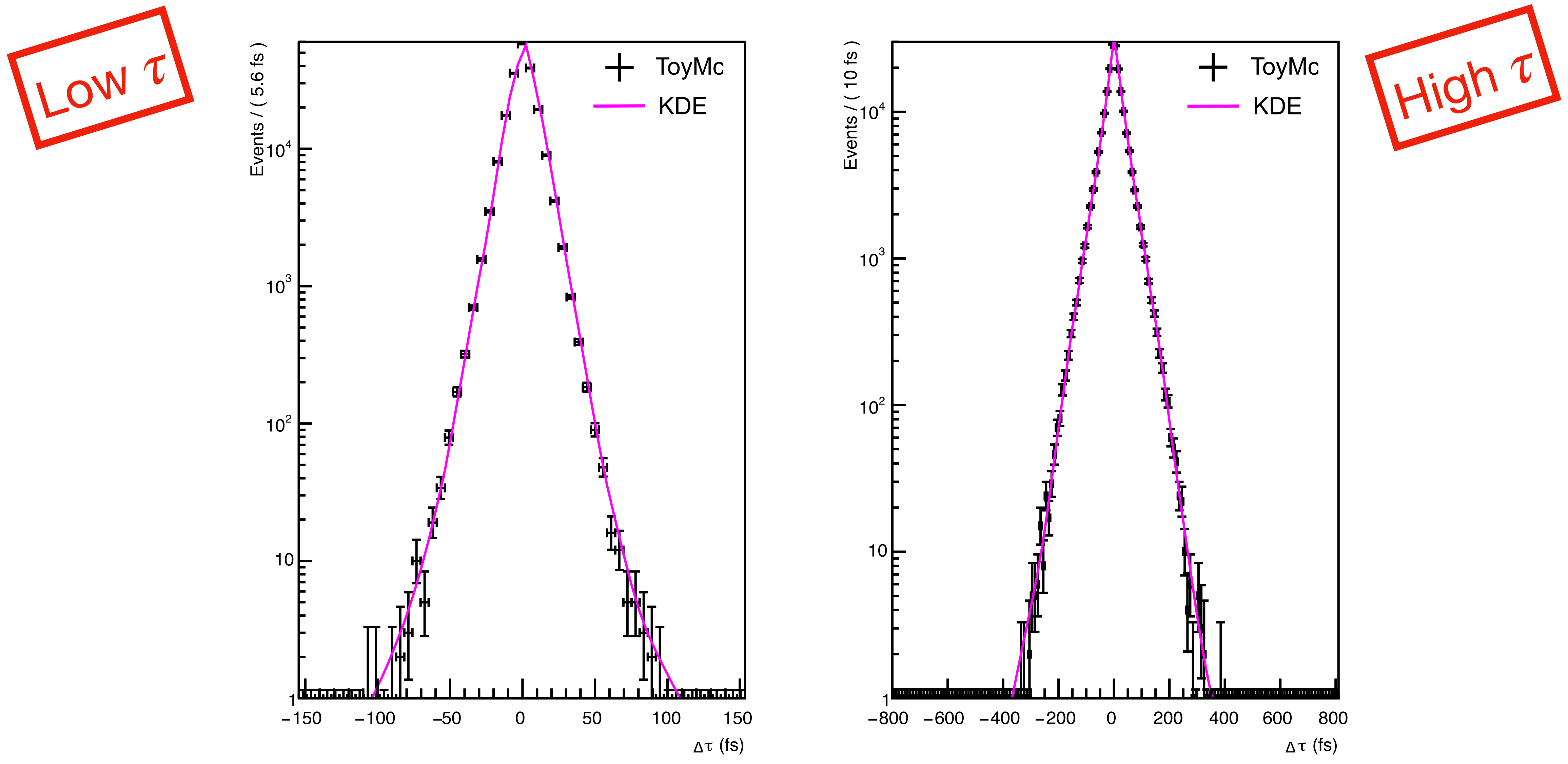
- $\sigma(p)/p = 0.5 - 1\,\%$
- $\sigma(IP) = (15 + 29/p_T)\mu m$
- $\sigma_{x,y,(z)}(PV) \sim 10(54)\mu m$
- 90% efficient $\pi - K$ separation
- Real-time calibration and alignment

Calculating the D^0 decay time

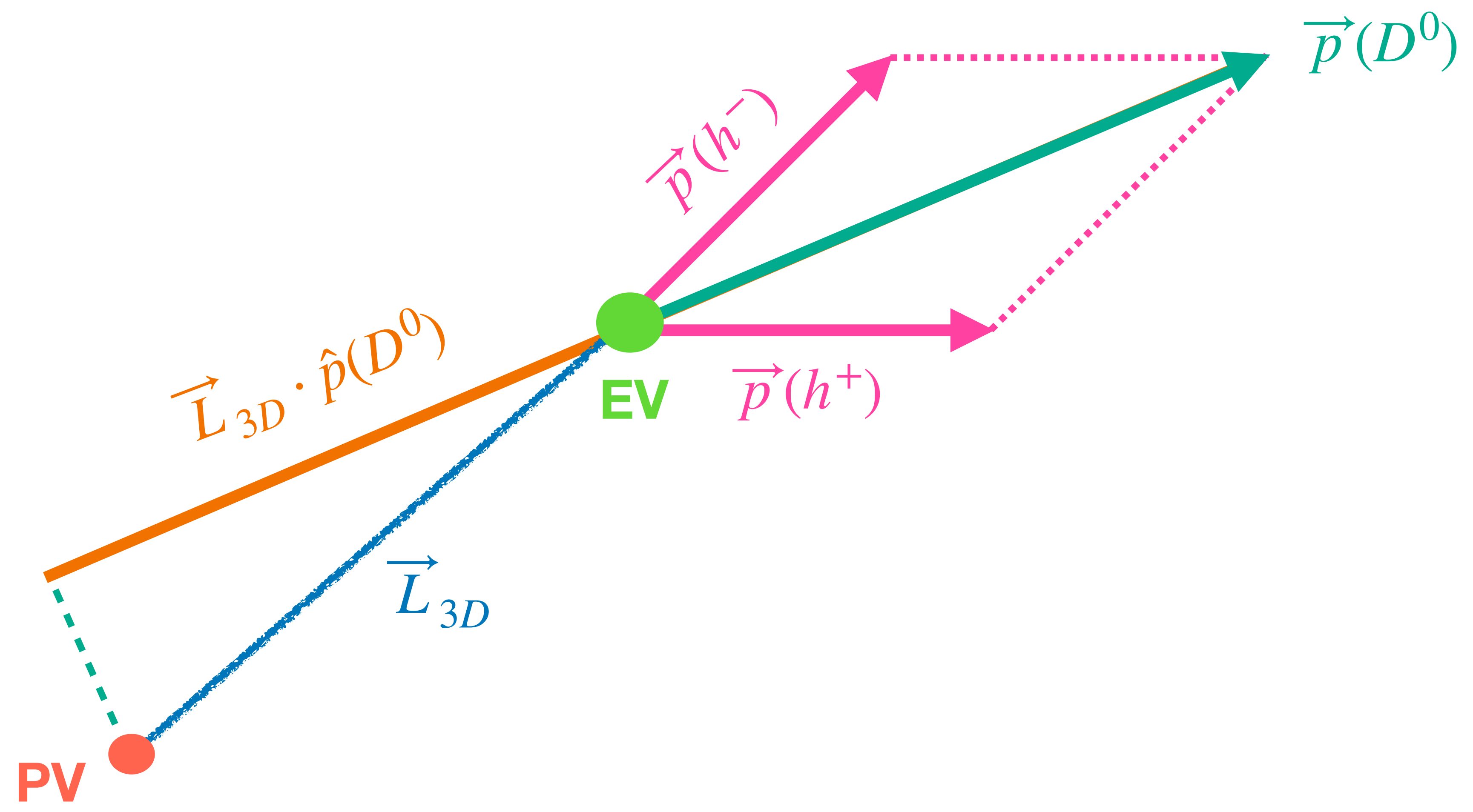
Variable resolution across the τ covered space

- In order to reduce the impact of resolution effects
  - projection of the flight distance onto the momentum

$$\tau = \frac{\overrightarrow{L} \cdot \vec{p}\, m}{\left|\vec{p}\,\right|^2 c}$$

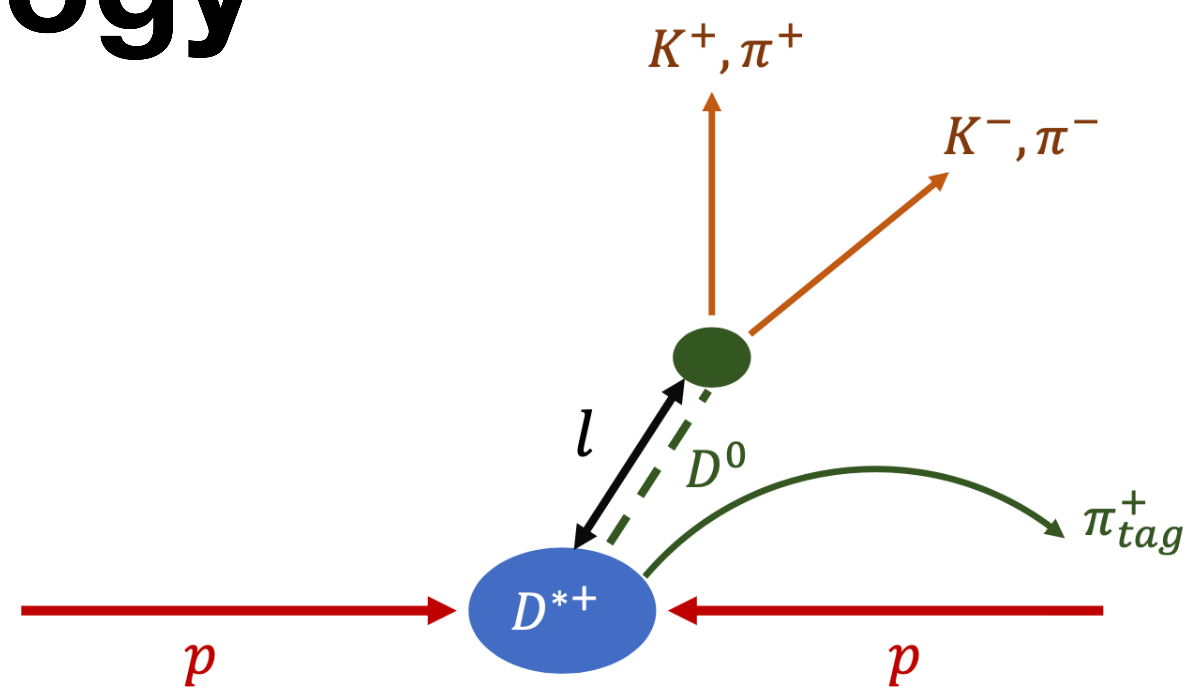


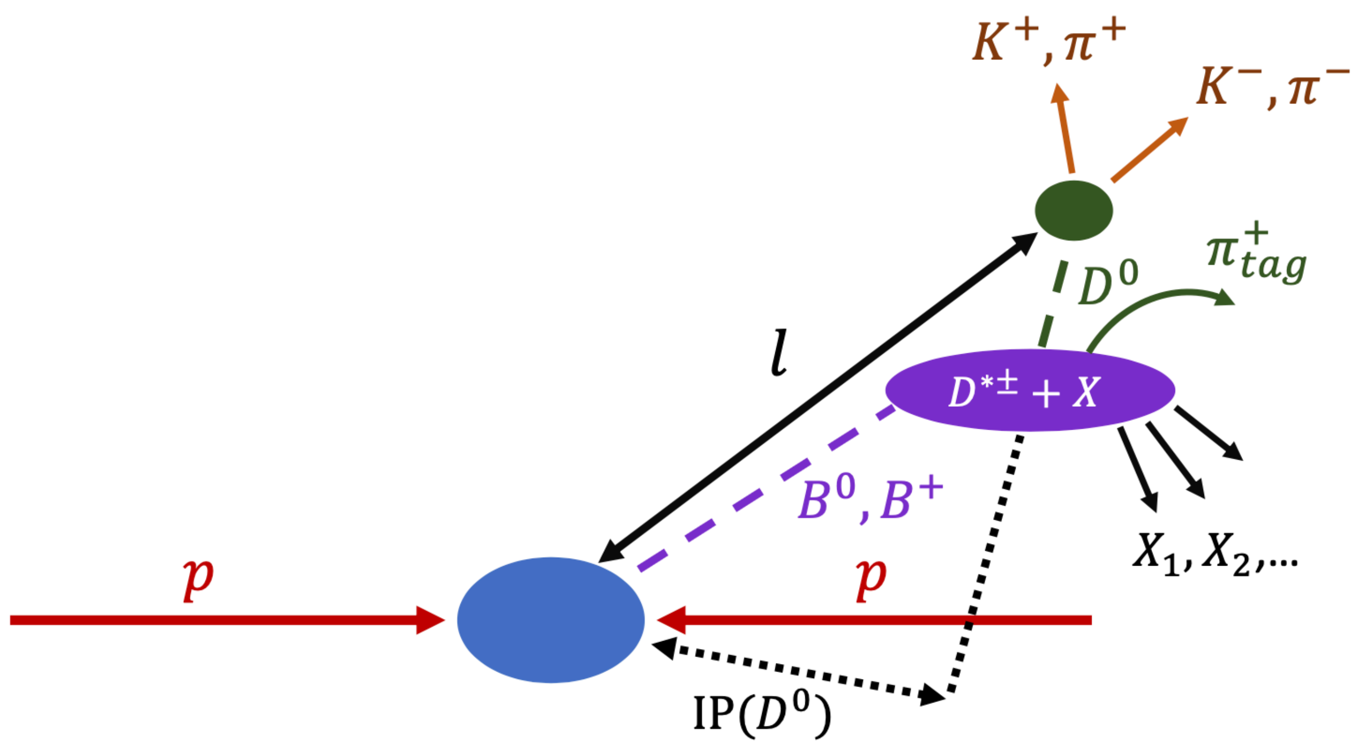
- Study of resolution template
  - Building a custom simulation (ToyMC): $3 \cdot 10^8$ events
  - Implementing finite resolution effects extracted from Official LHCb simulation
  - Simulated $\tau_{D^0} = 410.1\, fs$



Secondary decays: a source of systematic uncertainty

Phenomenology





- D^{*+} mesons can originate from $B^{0(\pm)}$ decays
- Neutral (charged) B has mean lifetime $\tau = 3.83(3.97)\tau_{D^0}$
- D^0 flight distance computed w.r.t. the primary vertex then the decay time is biased
- Handling secondary decays:
  - Selection on D^0 IP or χ^2_{IP} → still a fraction contaminates the sample and bias the extracted lifetime
  - Subtraction or inclusion in the fit → knowledge of secondary fraction in each decay time bin
- Due to the different topology the time dependent resolution is different form the one for prompt decays

Likelihood in case of variable resolution

A common mistake

- We want to extract f the fraction of event type A given a data sample containing type A and type B events

- In case of $p(x\,|\,A) = G(0,\sigma)$ and $p(x\,|\,B) = G(1,\sigma)$ the likelihood can be written as

$$\mathscr{L}(f;x) = \prod_{i=1}^{N} fG(x_i,1,\sigma) + (1-f)G(x_i,0,\sigma)\,. \qquad (1)$$

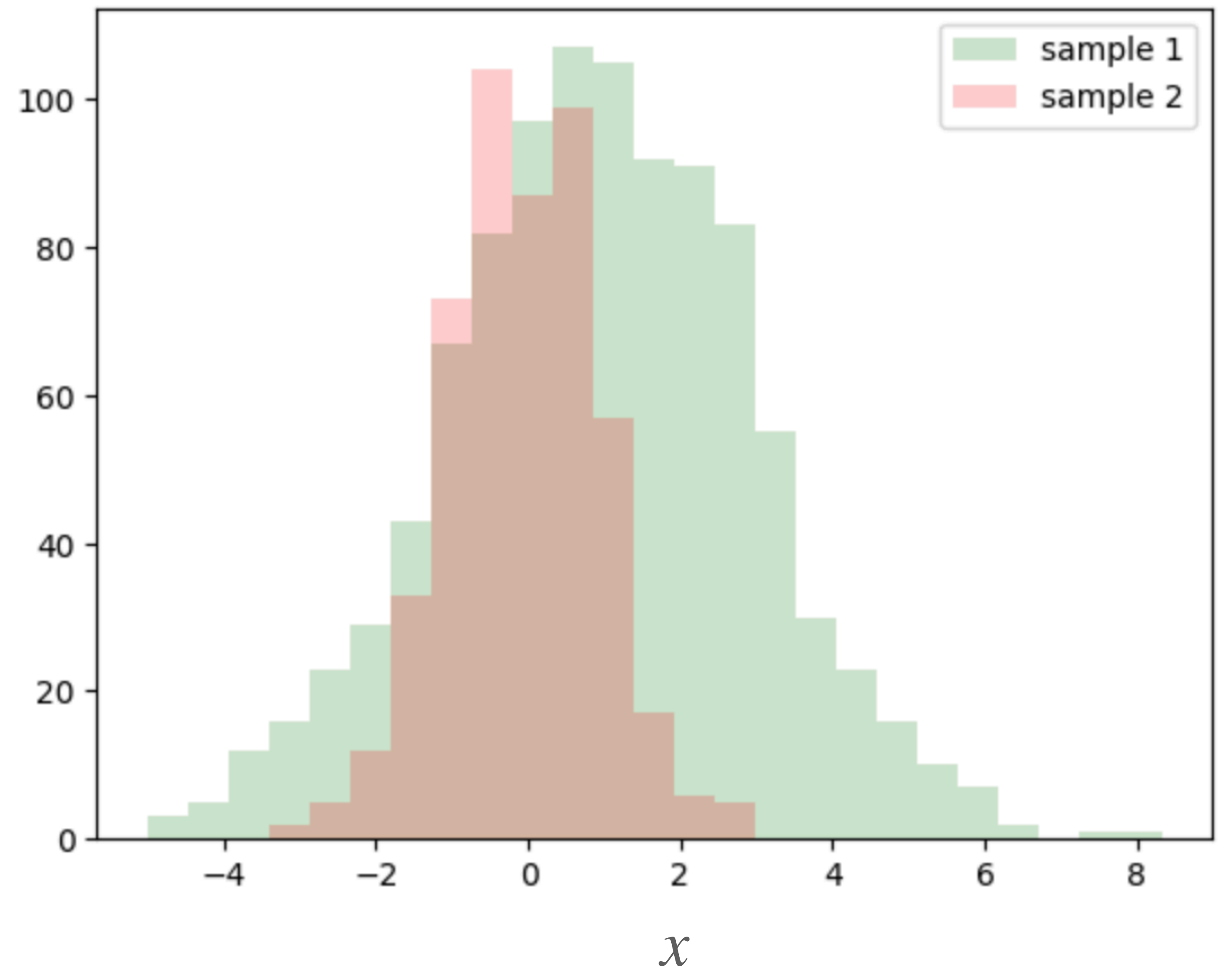
- In case of σ varying on event basis, one could be tempted to write

$$\mathscr{L}(f;x) = \prod_{i=1}^{N} fG(x_i,1,\sigma_i) + (1-f)G(x_i,0,\sigma_i)\,. \qquad (2)$$

- This leads to very dissatisfying results

Likelihood in case of variable resolution

Study of common mistake

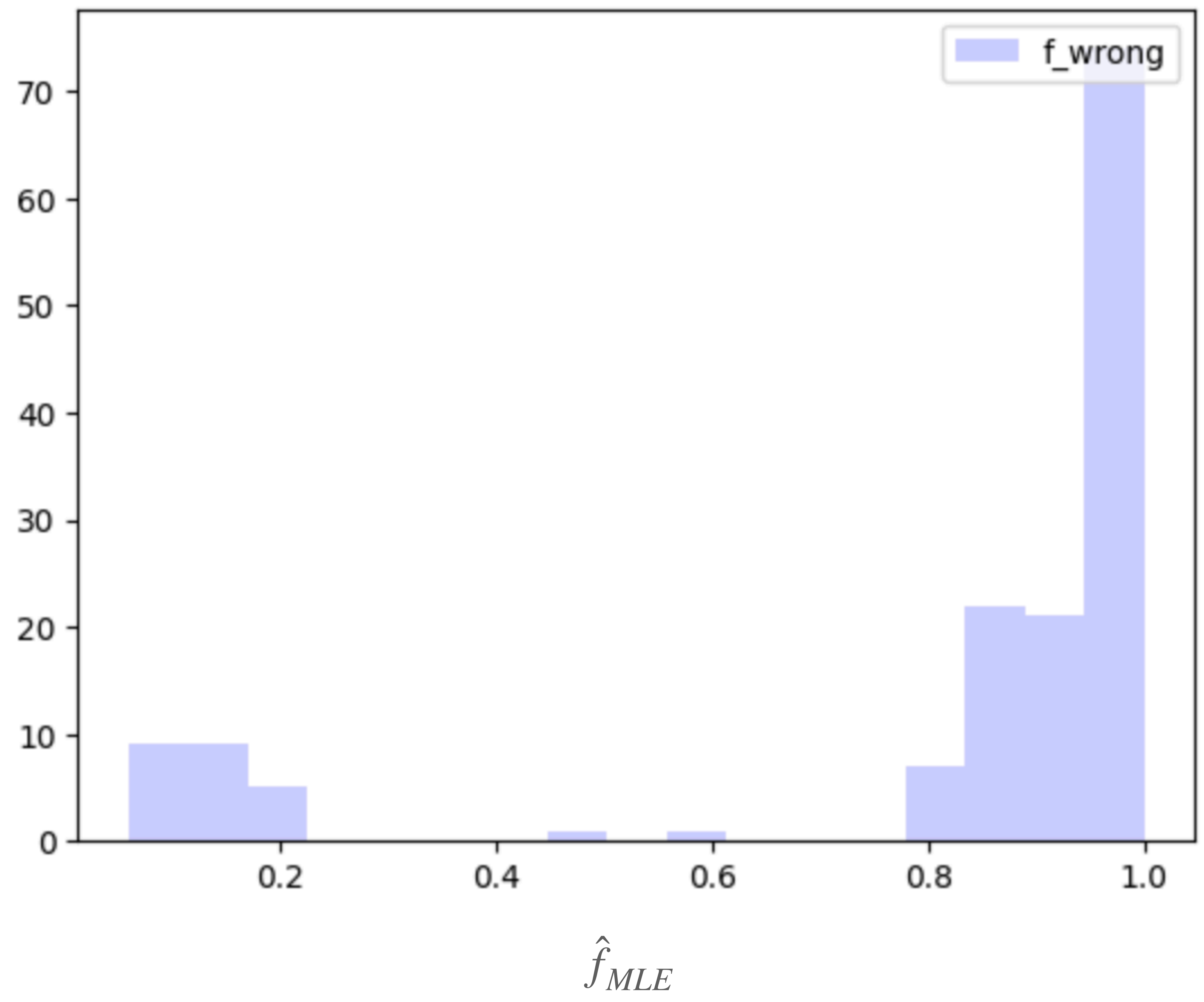


- Using a Jupyter notebook a Toy MC has been produced: link here
- A simpler case is studied for computational issue:
  - instead of σ_i varying event by event, it is fixed to a value but $\sigma_A \neq \sigma_B$
  - The likelihood $\mathscr{L}$ in (2) on prev. slide is used to perform unbinned MLE for f
- 150 pseudo experiments have been generated, with 150 events each
  - f is fixed to 1/3
- Once computed the log-likelihood, its maximum is searched in a numerical way through a scan for f in $[0,1]$
- Once extracted $\hat{f}_{MLE}$ for every pseudo experiment, its distribution is plotted

```
for n in range(150):
    logLL_wrong = np.full(100, 0)
    sample_1   = stats.norm.rvs(size = 50, loc = 0, scale = 1)
    sigma_1 = np.full(50, 1)
    sample_2   = stats.norm.rvs(size = 100, loc = 1, scale = 2)
    sigma_2 = np.full(100, 2)
    sample = np.concatenate((sample_1, sample_2))
    sigma = np.concatenate((sigma_1, sigma_2))
    for i in range(100):
        for s in range(150):
            logLL_wrong[i] +=  np.log(frac_space[i]*stats.norm.pdf(sample[s],\
            scale=sigma[s]) + (1-frac_space[i])*stats.norm.pdf(sample[s], loc=1, scale=sigma[s]))
    LLmax = max(logLL_wrong)
    logLL_wrong = logLL_wrong - LLmax
    fpos = np.where(logLL_wrong == 0)
    fraction_wrong.append(frac_space[fpos][-1])
```

Likelihood in case of variable resolution

Results of common mistake



- Several issues
  - This estimator is not consistent (consistency expected from MLE)
  - Degeneracy towards the extreme cases
  - The expected value has not been retrieved once

$$\mathscr{L}(f;x) = \prod_{i=1}^{N} fG(x_i,1,\sigma_i) + (1-f)G(x_i,0,\sigma_i) \qquad (2)$$

- What is happening?
  - It seems that each the two pieces of (2) are getting confused with the σ_i belonging to the other type

Likelihood in case of variable resolution

Recovering the common mistake

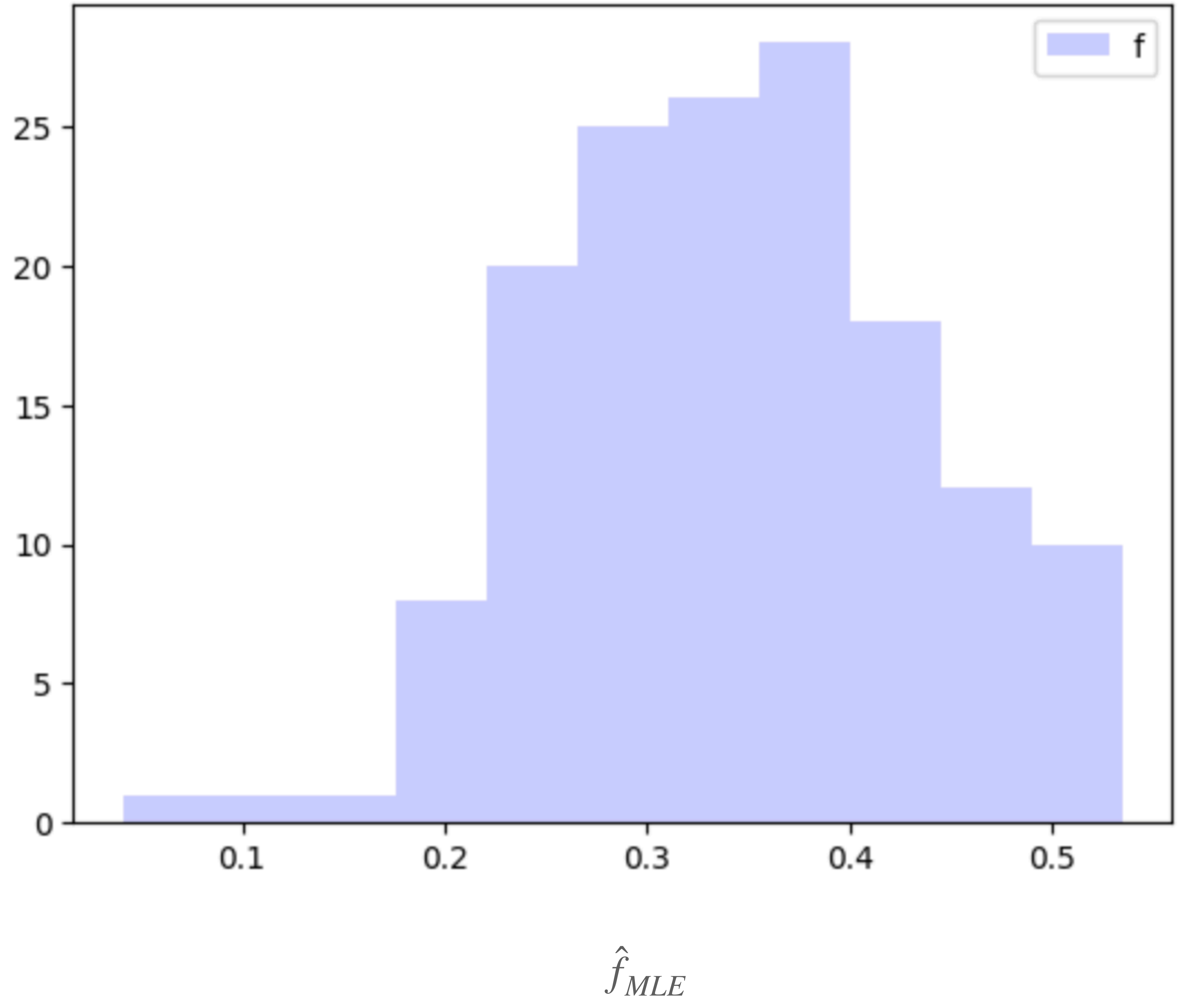
- The likelihood does not take in account the conditional probabilities
  - $p(\sigma_i|A)$ and $p(\sigma_i|B)$, hence the "confusion"
- In fact, the problem has two observables (x_i, σ_i), then the Likelihood must be written

$$\mathscr{L}(f;x) = \prod_{i=1}^{N} fp(x_i, \sigma_i|A) + (1-f)p(x_i \sigma_i|B)$$

and remembering $p(x_i, \sigma_i|X) = p(x_i|\sigma_i, X)p(\sigma_i|X)$, then

$$\mathscr{L}(f;x) = \prod_{i=1}^{N} fG(x_i, 1, \sigma_i)p(\sigma_i|A) + (1-f)G(x_i 0, \sigma_i)p(\sigma_i|B). \quad (3)$$

- It is found $<\hat{f}_{MLE}> = 0.344 \pm 0.007$



Likelihood in case of variable resolution

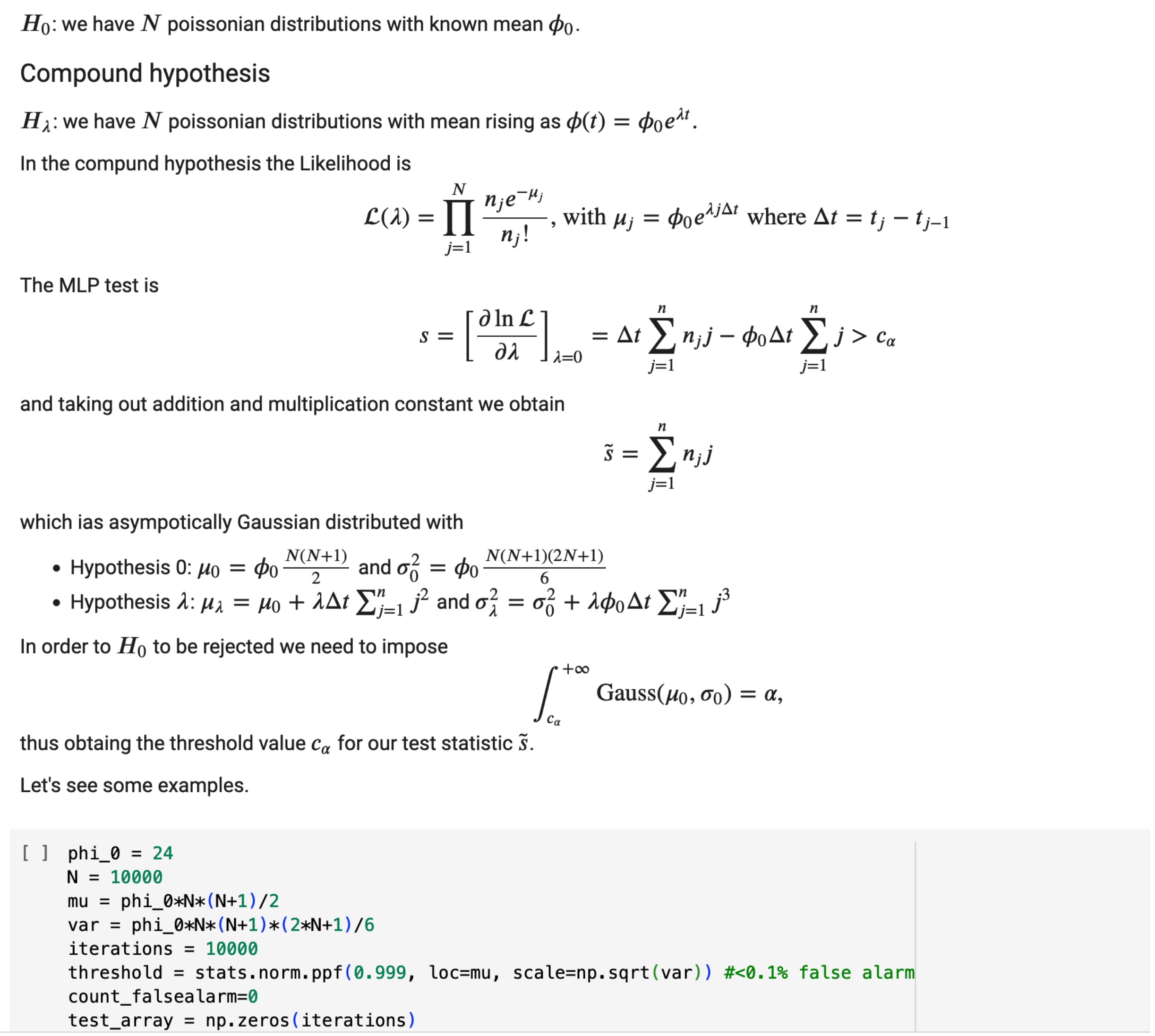
Conclusions

- In case of variable resolution in template fitting with different data type leading to a compound pdf
  - The full likelihood must be taken in account taking care of the distribution of σ_i itself within the classes
  - The penalty is a biased and inconsistent estimation
- When a simpler approach is allowed
  - From $\mathscr{L}(f;x) = \prod_{i=1}^{N} fG(x_i,1,\sigma_i)p(\sigma_i|A) + (1-f)G(x_i,0,\sigma_i)p(\sigma_i|B)$

    in case $p(\sigma_i|A)=p(\sigma_i|B)$ for every i, then it can be factorised out as it is just a multiplicative factor
    - Does not affect the Likelihood shape and its maximum
- After this consideration we can move on on the y_{CP} analysis (maybe even absolute lifetime in the far future) without incurring in fear of utterly mishandling this systematic uncertainty

A side note

Jupyter notebook



H_0: we have N poissonian distributions with known mean ϕ_0.

Compound hypothesis

H_λ: we have N poissonian distributions with mean rising as $\phi(t) = \phi_0 e^{\lambda t}$.

In the compund hypothesis the Likelihood is

$$\mathcal{L}(\lambda) = \prod_{j=1}^{N} \frac{n_j e^{-\mu_j}}{n_j!}, \text{ with } \mu_j = \phi_0 e^{\lambda j \Delta t} \text{ where } \Delta t = t_j - t_{j-1}$$

The MLP test is

$$s = \left[\frac{\partial \ln \mathcal{L}}{\partial \lambda}\right]_{\lambda=0} = \Delta t \sum_{j=1}^{n} n_j j - \phi_0 \Delta t \sum_{j=1}^{n} j > c_\alpha$$

and taking out addition and multiplication constant we obtain

$$\tilde{s} = \sum_{j=1}^{n} n_j j$$

which ias asympotically Gaussian distributed with

- Hypothesis 0: $\mu_0 = \phi_0 \frac{N(N+1)}{2}$ and $\sigma_0^2 = \phi_0 \frac{N(N+1)(2N+1)}{6}$
- Hypothesis λ: $\mu_\lambda = \mu_0 + \lambda \Delta t \sum_{j=1}^{n} j^2$ and $\sigma_\lambda^2 = \sigma_0^2 + \lambda \phi_0 \Delta t \sum_{j=1}^{n} j^3$

In order to H_0 to be rejected we need to impose

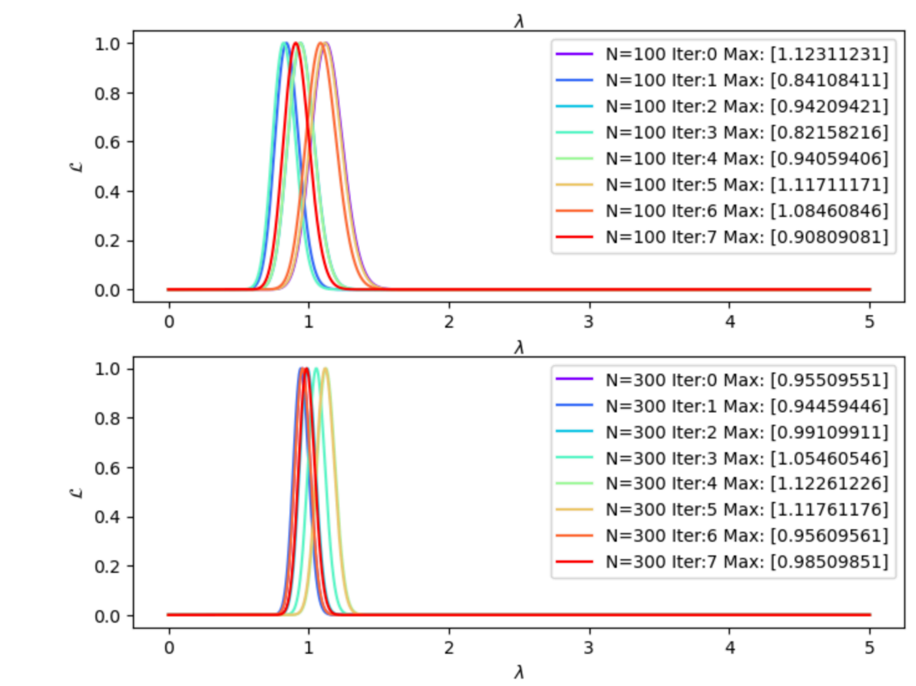
$$\int_{c_\alpha}^{+\infty} \text{Gauss}(\mu_0, \sigma_0) = \alpha,$$

thus obtaing the threshold value c_α for our test statistic $\tilde{s}$.

Let's see some examples.

```
[ ] phi_0 = 24
    N = 10000
    mu = phi_0*N*(N+1)/2
    var = phi_0*N*(N+1)*(2*N+1)/6
    iterations = 10000
    threshold = stats.norm.ppf(0.999, loc=mu, scale=np.sqrt(var)) #<0.1% false alarm
    count_falsealarm=0
    test_array = np.zeros(iterations)
```

- This study is based on ref. [1]
  - The decision to rerun a similar study on Jupyter Notebooks is mainly due to
    - Faster and deeper understanding of the phenomenon as one can “play” with the case and parameters
    - Didactical purpose: such notebooks can be reproposed to students for exercise
    - Side study of numerical approximation and computation of point/interval estimation using various tool
  - Available other notebook:
    - MLE on exponential
    - Confidence Intervals on exponential
    - Hypothesis testing

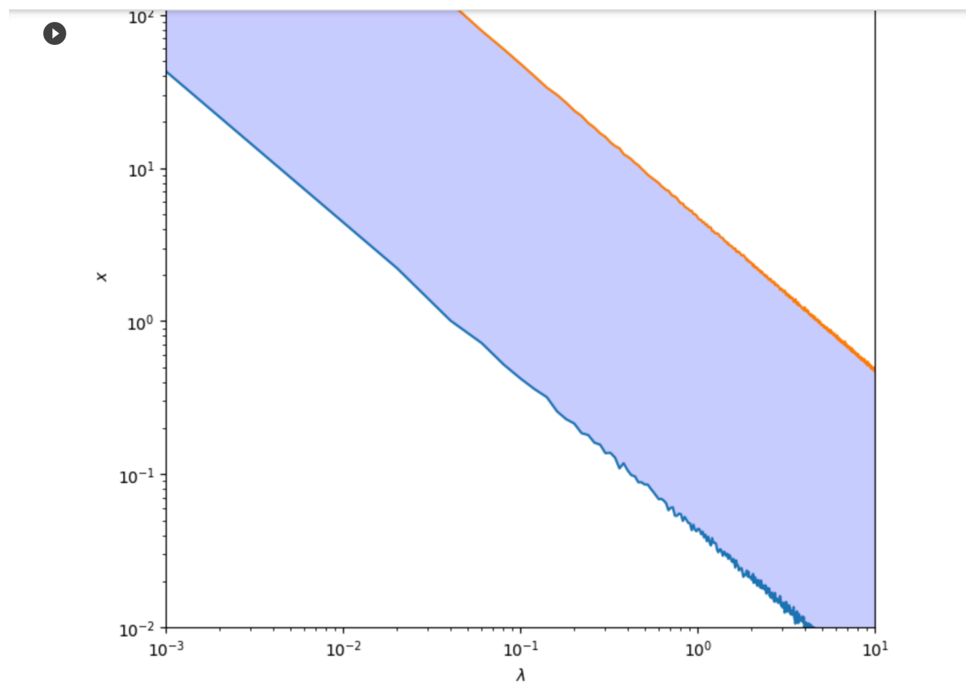


The MLE for λ however is a biased estimator as it can be proven that

$$E\left[\hat{\lambda}_{MLE} - \lambda\right] = E\left[\hat{\lambda}_{MLE} - \frac{N}{\sum_{i=1}^{N} x_i}\right] = \frac{\lambda}{N-1},$$

however a simple transformation $\hat{\lambda}_{MLE} \rightarrow \frac{N-1}{N}\hat{\lambda}_{MLE}$ will give us an unbiased estimator as $E\left[\frac{N-1}{N}\hat{\lambda}_{MLE} - \lambda\right] = 0$.

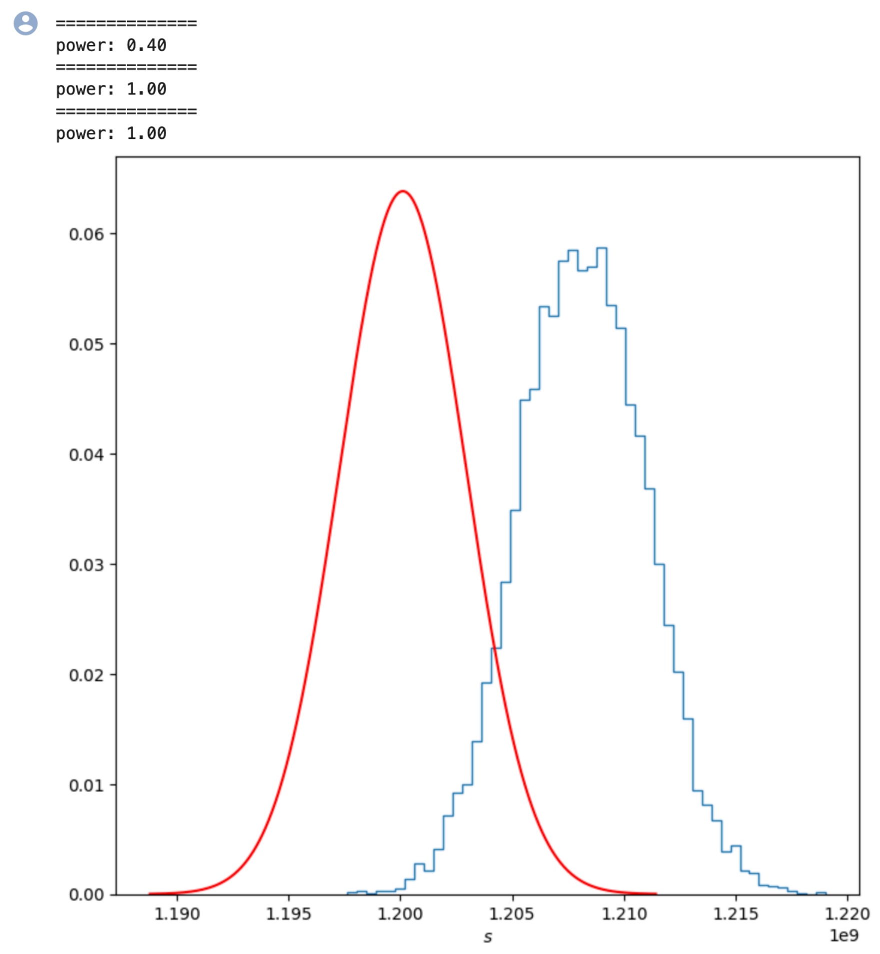
Also with non-linear transformation $\lambda \rightarrow \frac{1}{\theta}$, we can reparemetrize the likelihhod as



Coverage (upper limit)

```
[ ] def compute_upper_limit(val, upper_band, par):
        min_distance = np.abs(np.array(upper_band - val))
        return par[np.argmin(min_distance)]

    iterations = 20000
```



References

[1] G. Punzi, Comments on Likelihood Fits with Variable Resolution, PHYSTAT2003, SLAC, Stanford, California, September 8-11, 2003

[2] F. Terzuoli, Precision measurement of time-dependent charm observables at high-luminosity LHCb, CERN-THESIS-2022-029

[3] F. James, Statistical Methods for Experimental Physics (2nd edition), World Scientific Publishing, 2006

[4] Jupyter Notebook: Likelihood fits with variable resolution