

# GEANT4/NPTOOL 시뮬레이션 – 2/2

이정우



# 예제 (ionization chamber)

lilak update



# 목표

1. Ionization chamber 에서 입자가 잃어버리는 에너지를 재어보자.
2. 두가지 입자가 섞였을 때 두 입자를 구분해 보자.

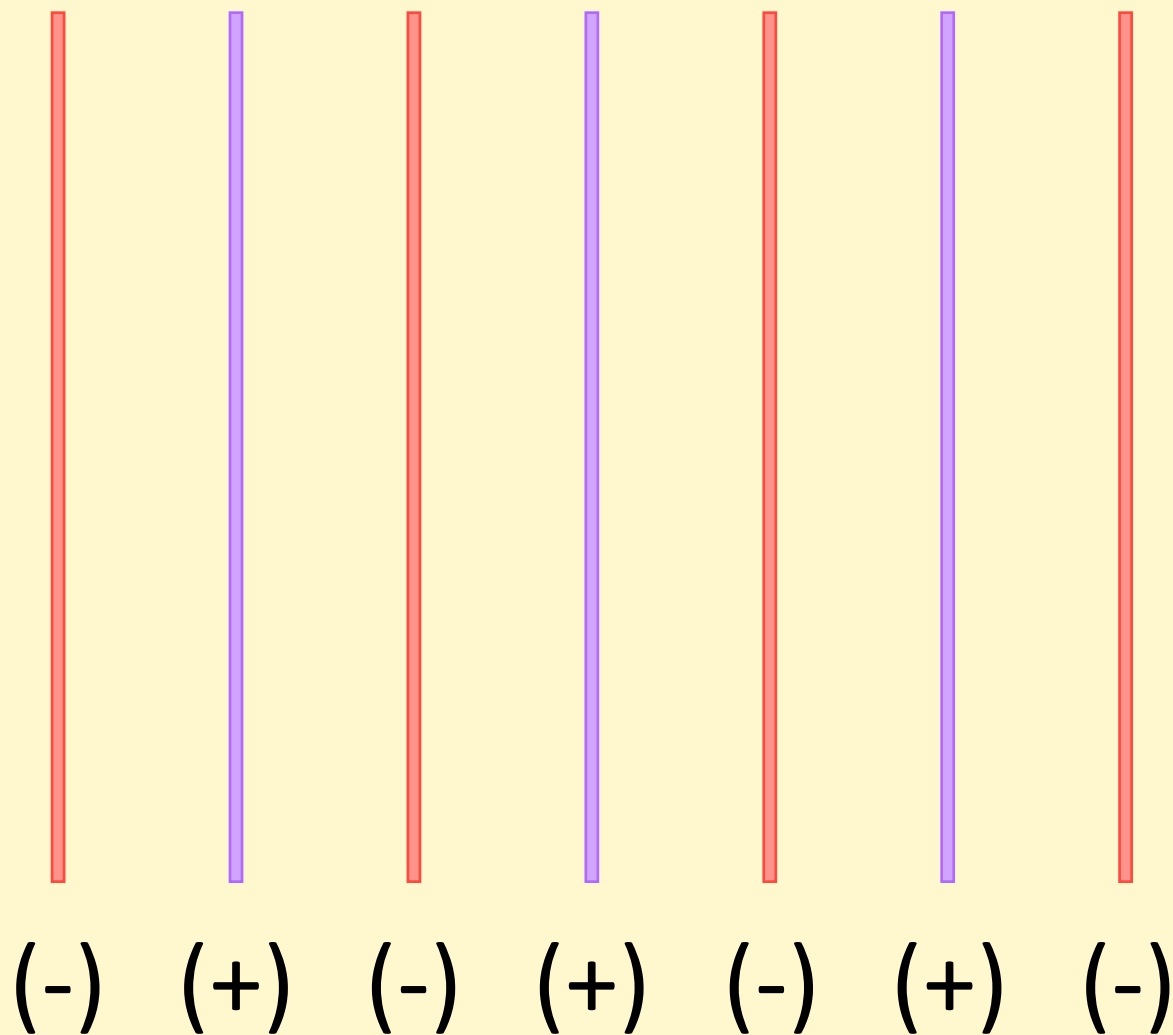
# 시뮬레이션에서 고려해야 할 것

1. 원하는 반응에서 나오는 입자들
2. 원하는 반응이 아닌 반응에서 나오는 입자들
3. 입자의 방향 및 에너지 정확도
4. 검출기
5. 검출기 위치 및 크기 정확도
6. 검출기 프레임 및 환경
7. 검출기 입자 검출 과정
8. 데이터 전송 과정
9. 검출기 분해능

# 시뮬레이션에서 고려해야 할 것

1. 원하는 반응에서 나오는 입자들
- ~~2. 원하는 반응이 아닌 반응에서 나오는 입자들~~
3. 입자의 방향 및 에너지 정확도
4. 검출기
- ~~5. 검출기 위치 및 크기 정확도~~
- ~~6. 검출기 프레임 및 환경~~
- ~~7. 검출기 입자 검출 과정~~
- ~~8. 데이터 전송 과정~~
9. 검출기 분해능

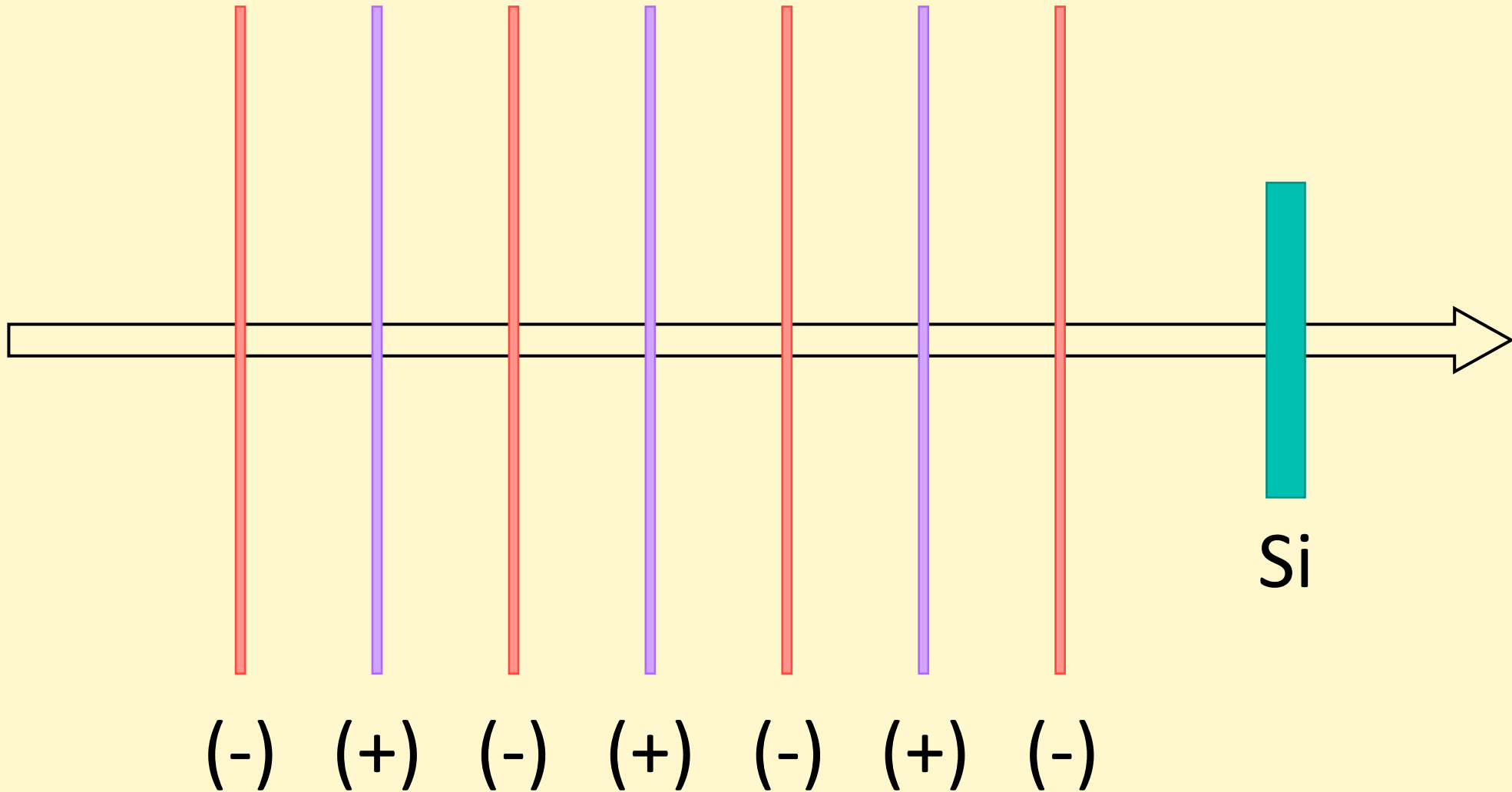
# IONIZATION CHAMBER



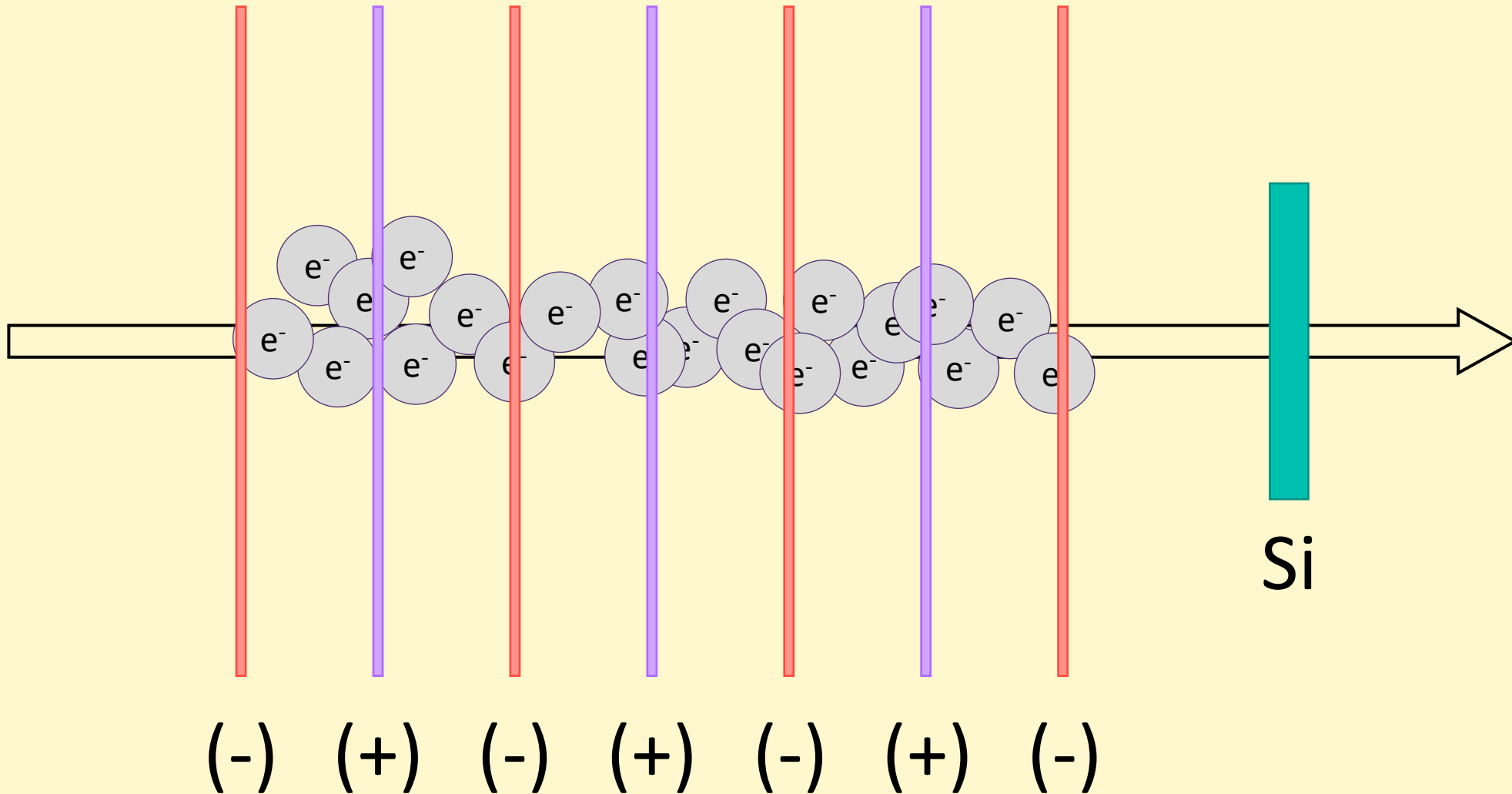
Si



# IONIZATION CHAMBER

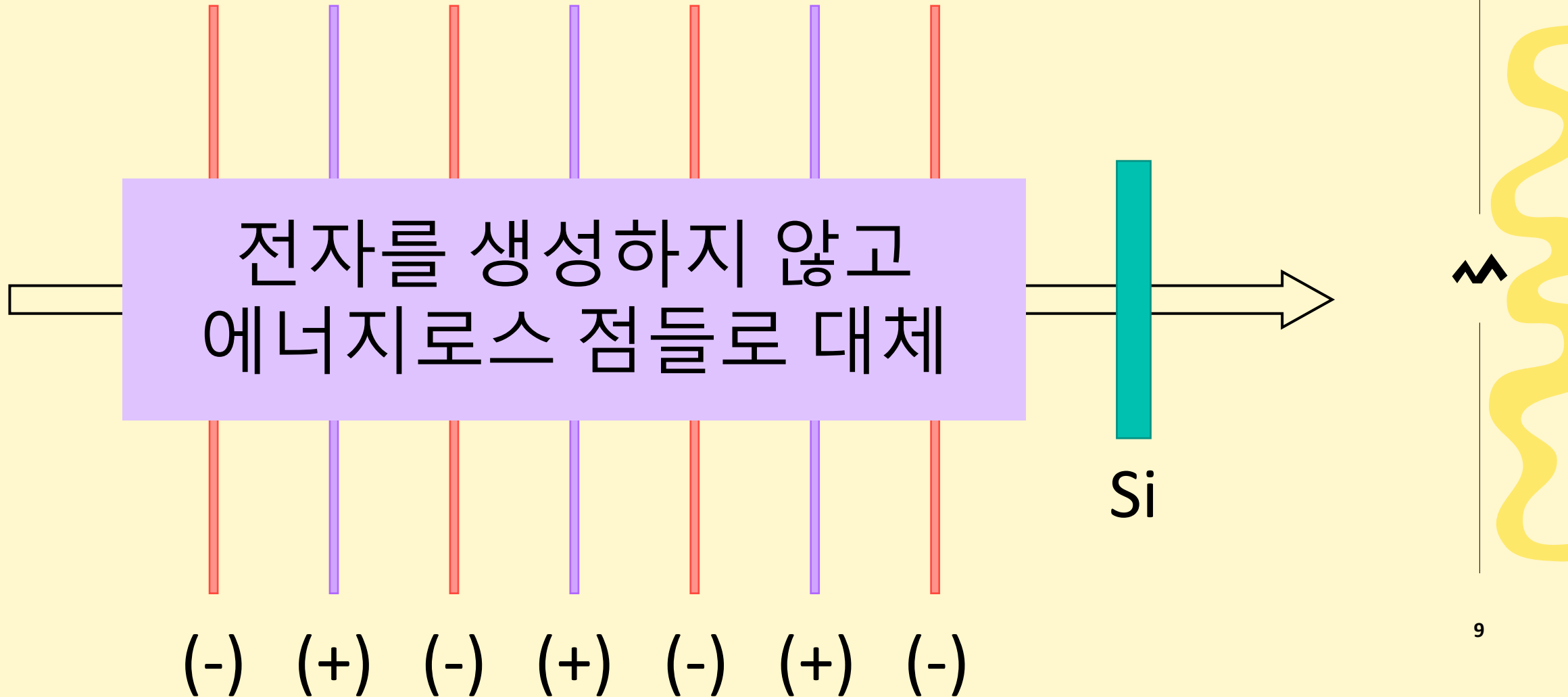


# IONIZATION CHAMBER

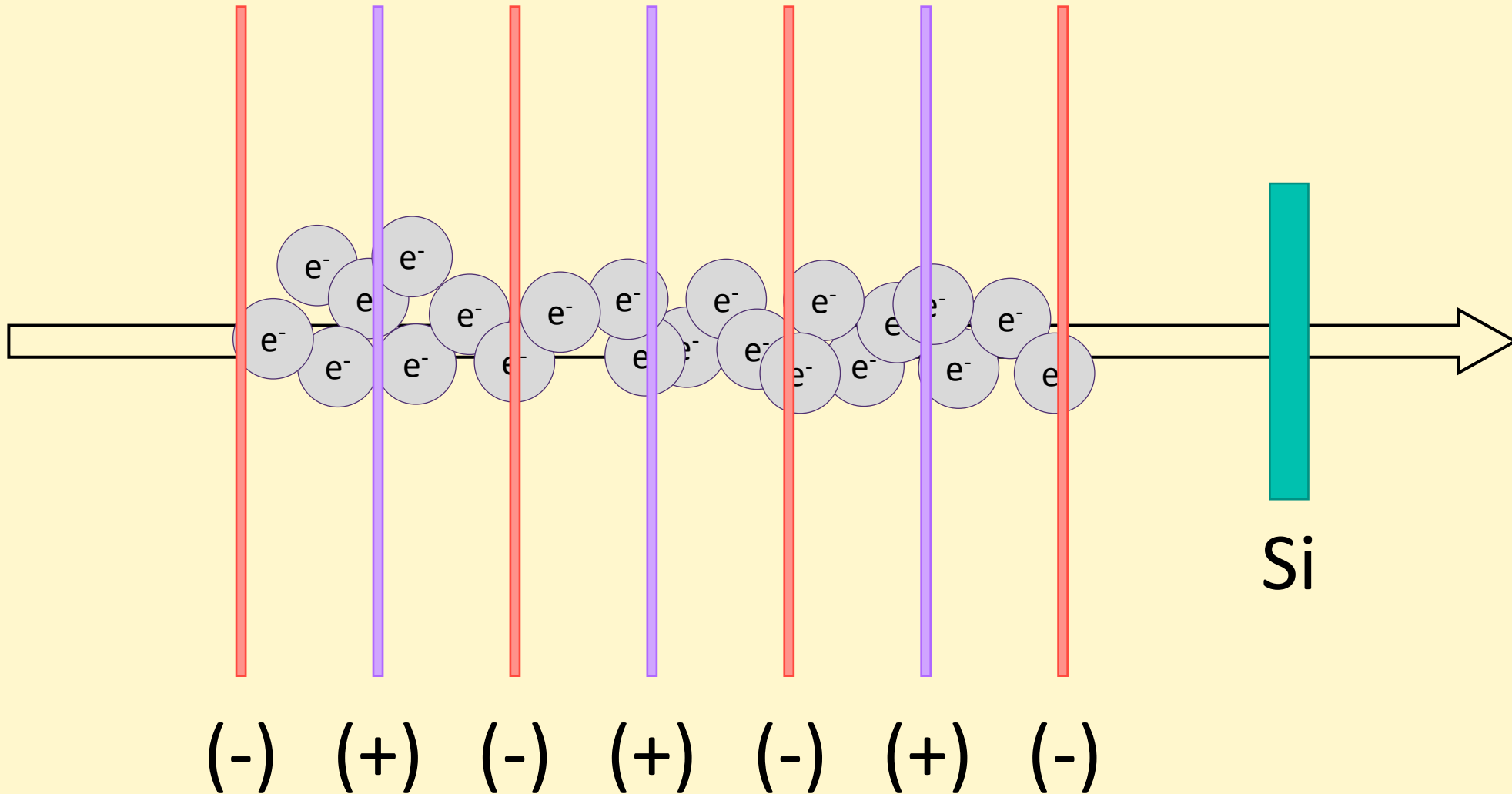




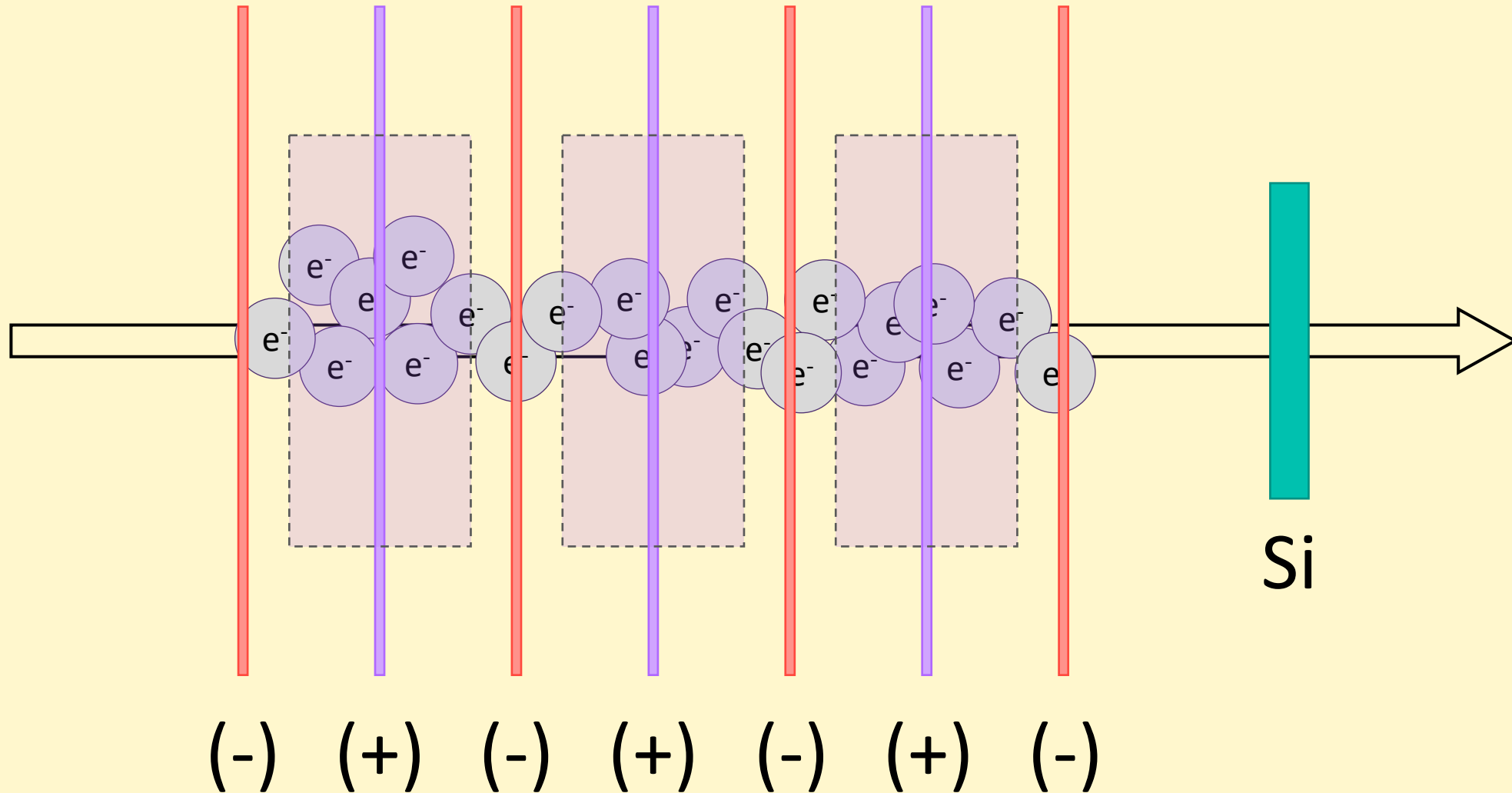
# IONIZATION CHAMBER



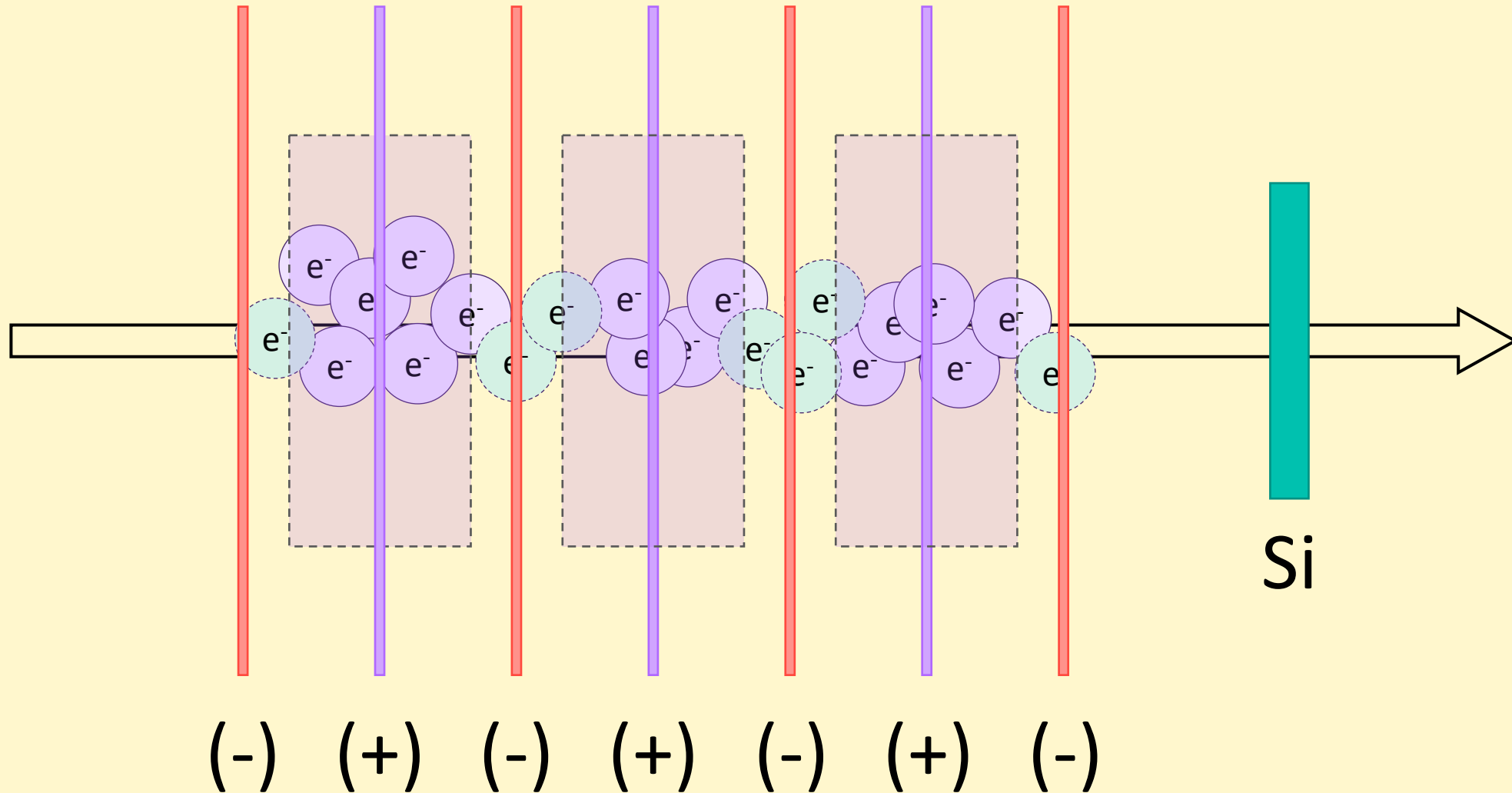
# IONIZATION CHAMBER



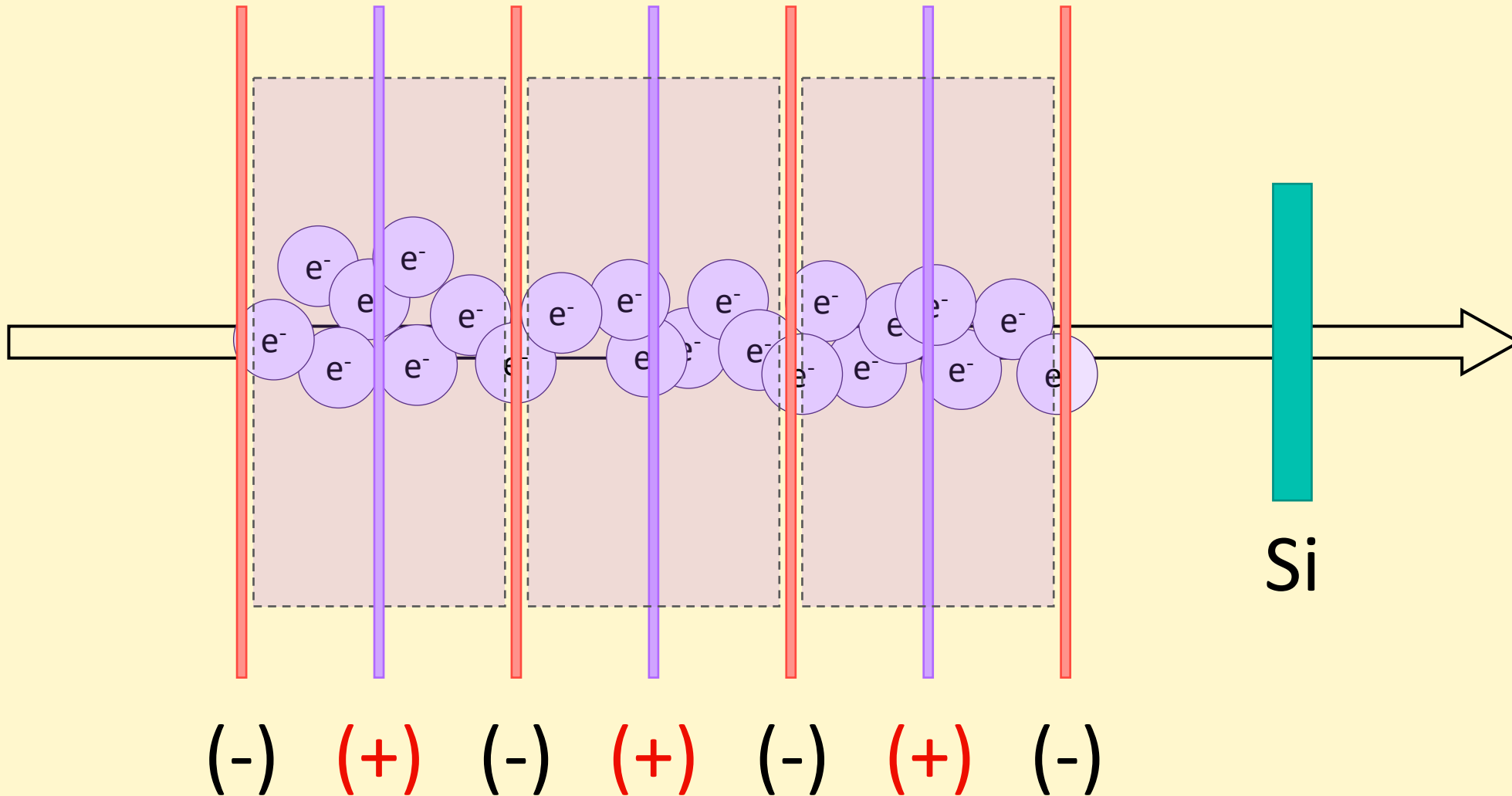
# IONIZATION CHAMBER



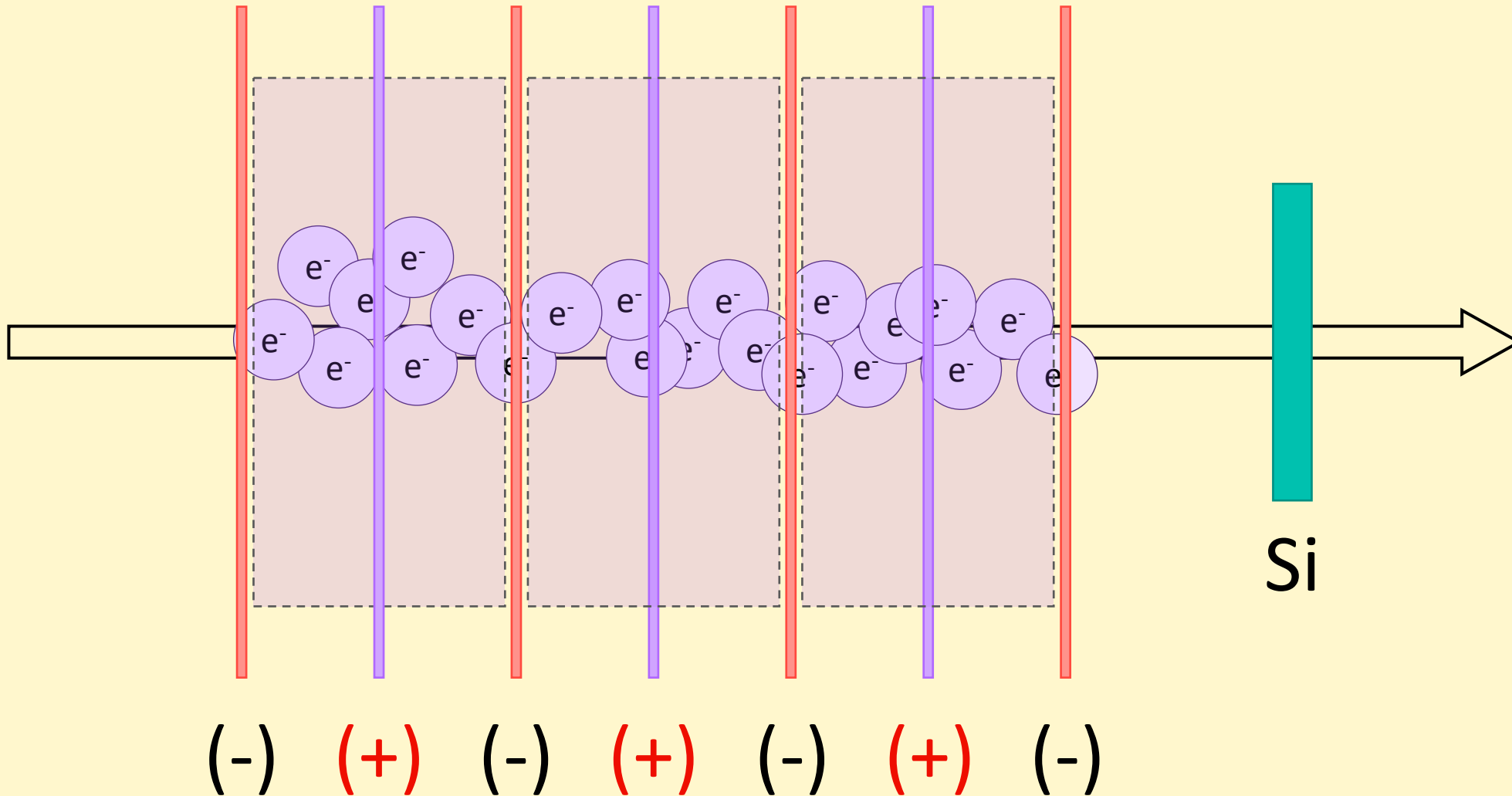
# IONIZATION CHAMBER



# IONIZATION CHAMBER



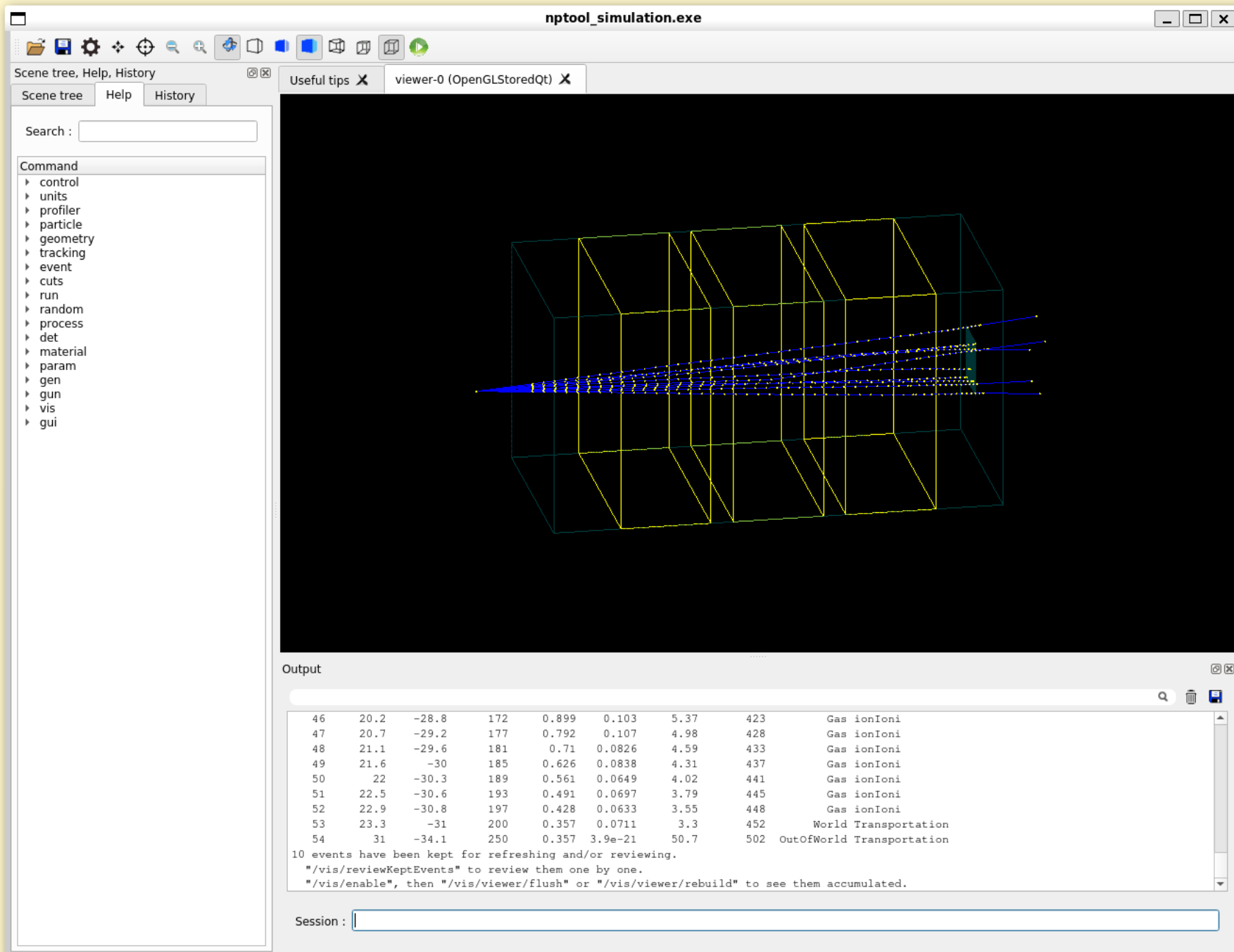
# IONIZATION CHAMBER



# 시뮬레이션 설정

1. `configure_alpha_source.mac` : 5.5 MeV  $^4\text{He}$  source (5 deg)
2. `configure_ranged_e_alpha.mac` : 2 – 8 MeV  $^4\text{He}$  beam (20 deg)
3. `configure_ranged_e_3He.mac` : 2 – 8 MeV  $^3\text{He}$  beam (20 deg)

lilak nptool `configure_alpha_source.mac`





# CONFIGURE\_ALPHA\_SOURCE.MAC

```
#nbeams 10000  
#verbose 0  
#use_vis false  
#dLoss 10
```

```
nbeams 1  
verbose 1  
use_vis true  
dLoss 10
```

## CSSUDetectorConstruction/

```
GasMaterial      CF4 # gas material name.  
GasFraction      1  
GasPressure      30 torr # bar or torr  
GasTemperature   295 # kelvin  
WorldSize        200., 200., 500. # mm  
GasVolumeSize    200., 200., 400. # mm  
SensitiveRanges  50+{dLoss}, 150-{dLoss}, 150+{dLoss}, 250-{dLoss}, 250+{dLoss}, 350-{dLoss} # mm
```

# CONFIGURE\_ALPHA\_SOURCE.MAC

```
#nbeams 10000  
#verbose 0  
#use_vis false  
#dLoss 10
```

} 데이터 생성용 (그림x 빔10000)

```
nbeams 1  
verbose 1  
use_vis true  
dLoss 10
```

} 확인용 (그림o, 빔 1개)

CSSUDetectorConstruction/

```
GasMaterial      CF4 # gas material name.  
GasFraction      1  
GasPressure      30 torr # bar or torr  
GasTemperature   295 # kelvin  
WorldSize        200., 200., 500. # mm  
GasVolumeSize    200., 200., 400. # mm  
SensitiveRanges  50+{dLoss}, 150-{dLoss}, 150+{dLoss}, 250-{dLoss}, 250+{dLoss}, 350-{dLoss} # mm
```

# 를 줄 앞에 붙이면 명령어가  
적용이 안됨 (comment out)

# 데이터 생성



# 데이터 생성

- configuration\_alpha\_source.mac 파일을 열어서
- 그림 + 확인용 명령줄을 comment out 하고 (#을 추가)  
데이터 생성용 명령줄을 다시 적용 (#을 제거) 해보자.
- 시뮬레이션 실행
- lilak nptool configuration\_alpha\_source.mac
- **ls** data/cssu\_4He\_5.5\_5.5\_5deg\_CF4\_30torr\_d10\_10000.info
- **ls** data/cssu\_4He\_5.5\_5.5\_5deg\_CF4\_30torr\_d10\_10000.dat

**cat** data/cssu\_4He\_5.5\_5.5\_5deg\_CF4\_30torr\_d10\_10000.info

n\_events 10000

n\_branches 4

branch 0 MCStepSensitiveBlock\_60\_140

branch 1 MCStepSensitiveBlock\_160\_240

branch 2 MCStepSensitiveBlock\_260\_340

branch 3 MCStepSi

event	branch_id	step_id	track_id	x	y	z	t	e
-------	-----------	---------	----------	---	---	---	---	---

# 분석 매크로

1. `analysis_header.h` : 기본 설정들
2. `analysis_1_read_data.C` : 각 검출 영역에 대한 에너지 분포
3. `analysis_2_fit_data.C` : 에너지 분포 피팅
4. `analysis_3_ee.C` : 에너지1 vs 에너지2 분포
5. `analysis_4_pid.C` : 에너지1 vs 에너지2 분포 → PID

# cat analysis\_header.h

```
#ifndef analysis_header_h
#define analysis_header_h

TString default_file_name1 = "data/cssu_4He_5.5_5.5_5deg_CF4_30torr_d10_10000.dat";
//TString default_file_name1 = "data/cssu_3He_8_3_20deg_CF4_30torr_d10_10000.dat";
//TString default_file_name1 = "data/cssu_4He_8_3_20deg_CF4_30torr_d10_10000.dat";

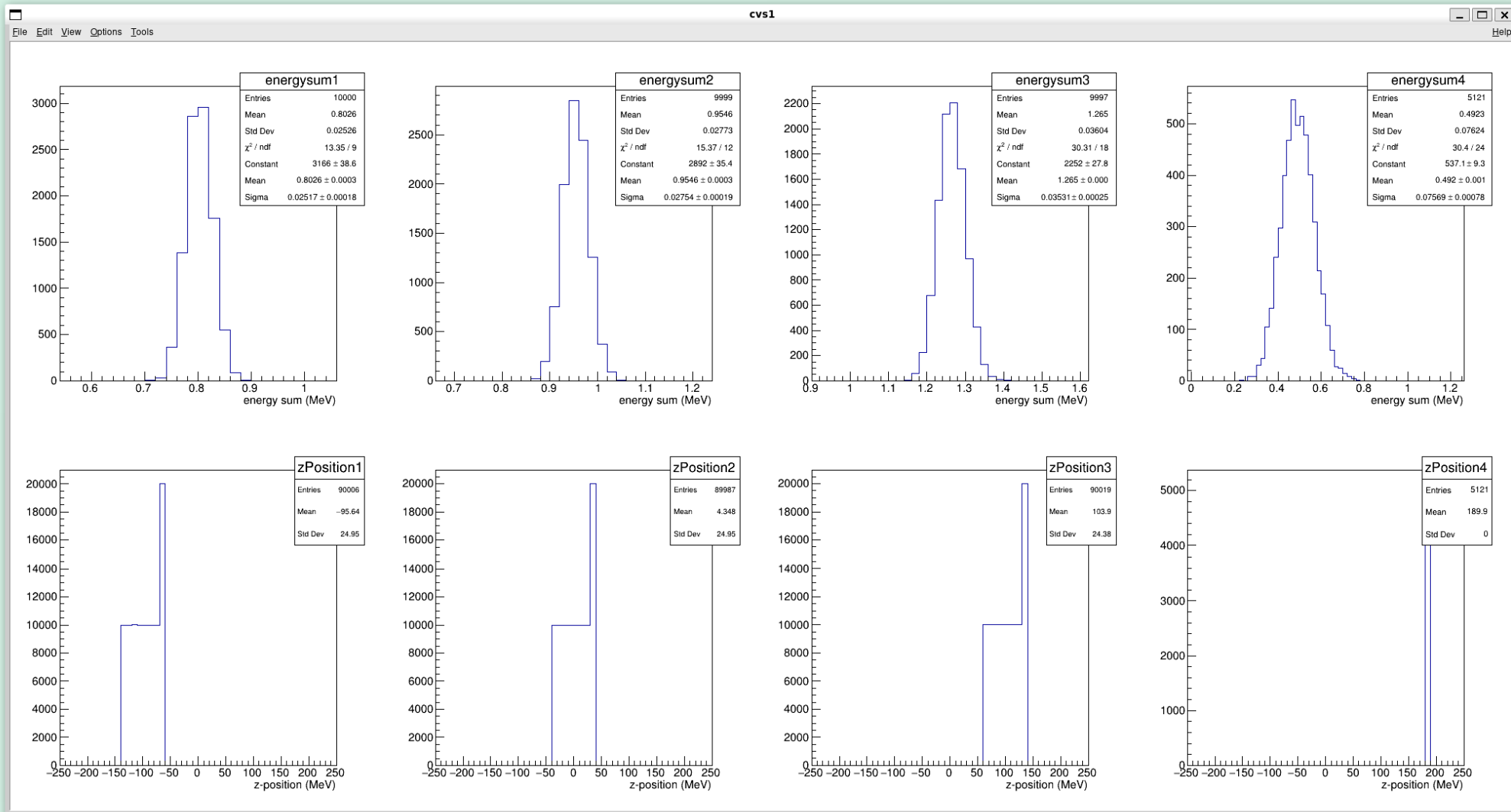
TString default_file_name2 = "";
//TString default_file_name2 = "data/cssu_3He_8_3_20deg_CF4_30torr_d10_10000.dat";

double energy_resolution = 0.01; // =sigma/mean

#endif
```

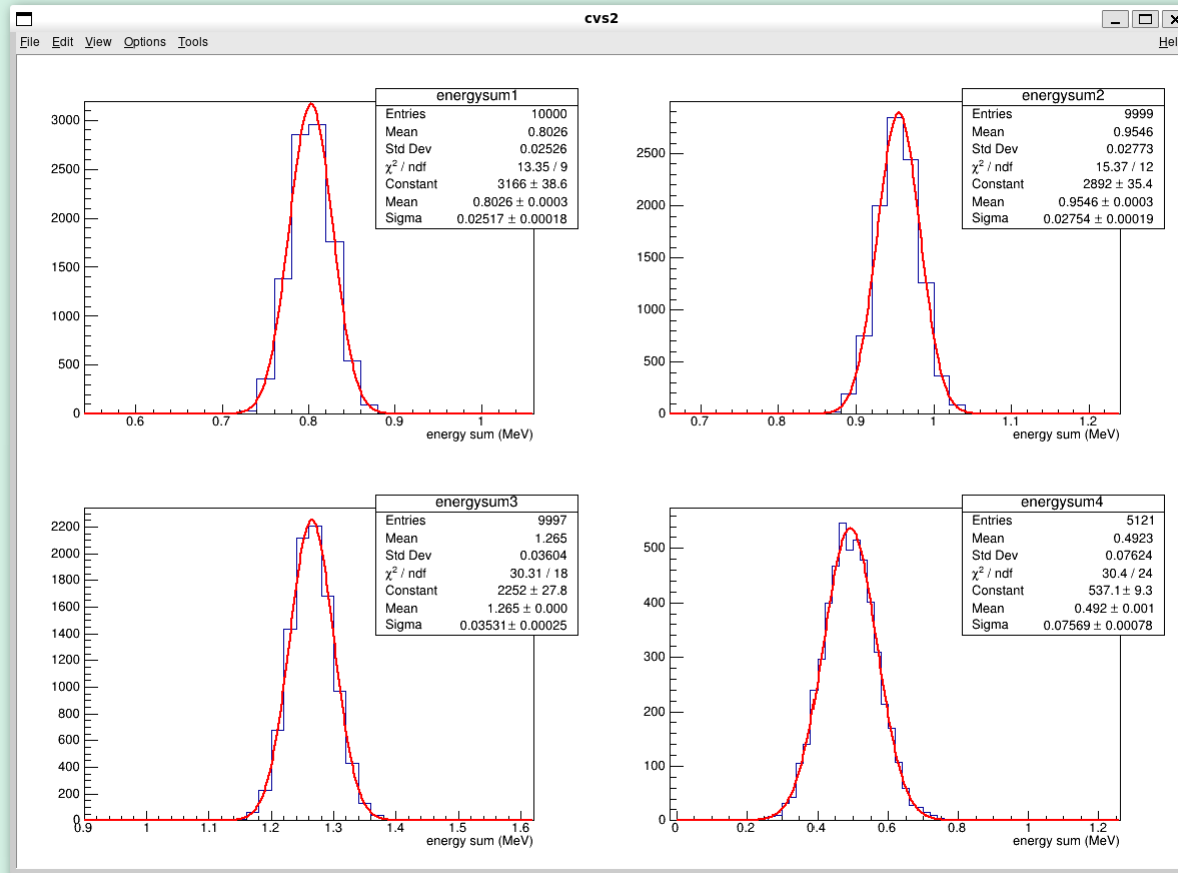
// 를 줄 앞에 붙여서 comment out

# root analysis\_1\_fit\_data.C





# root analysis\_2\_.C



이벤트마다 각 영역에 잃어버린 에너지를 채운 히스토그램 분포

# PID

## (particle Identification)



# 여러 입자를 섞어보자!

1. `analysis_header.h` : 기본 설정들
2. `analysis_1_read_data.C` : 각 검출 영역에 대한 에너지 분포
3. `analysis_2_fit_data.C` : 에너지 분포 피팅
4. `analysis_3_ee.C` : 에너지1 vs 에너지2 분포
5. `analysis_4_pid.C` : 에너지1 vs 에너지2 분포 → PID

실험을 하는데 두가지의 다른 입자가 나왔다! ( $^4\text{He}$ ,  $^3\text{He}$ )

# cat analysis\_header.h

```
#ifndef analysis_header_h
#define analysis_header_h

//TString default_file_name1 = "data/cssu_4He_5.5_5.5_5deg_CF4_30torr_d10_10000.dat";
//TString default_file_name1 = "data/cssu_3He_8_3_20deg_CF4_30torr_d10_10000.dat";
TString default_file_name1 = "data/cssu_4He_8_3_20deg_CF4_30torr_d10_10000.dat";

//TString default_file_name2 = "";
TString default_file_name2 = "data/cssu_3He_8_3_20deg_CF4_30torr_d10_10000.dat";

double energy_resolution = 0.01; // =sigma/mean

#endif
```

// 를 줄 앞에 붙여서 comment out

# 데이터 생성

- lilak nptool configuration\_ranged\_e\_alpha.mac
- lilak nptool configuration\_ranged\_e\_3He.mac

# root

```
#ifndef analysis_header_h
#define analysis_header_h

//TString default_file_name1 = "data/cssu_4He_5.5_5.5_5deg_CF4_30torr_d10_10000.dat";
//TString default_file_name1 = "data/cssu_3He_8_3_20deg_CF4_30torr_d10_10000.dat";
TString default_file_name1 = "data/cssu_4He_8_3_20deg_CF4_30torr_d10_10000.dat";

//TString default_file_name2 = "";
TString default_file_name2 = "data/cssu_3He_8_3_20deg_CF4_30torr_d10_10000.dat";

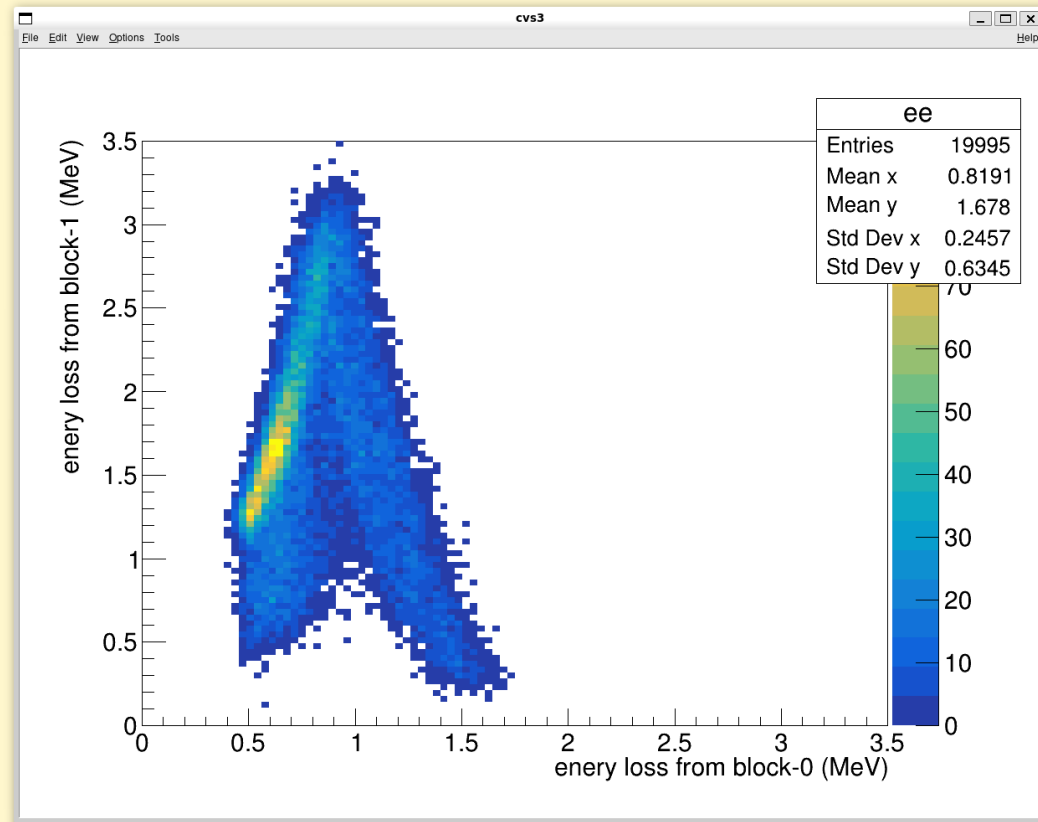
double energy_resolution = 0.01; // =sigma/mean

#endif
```

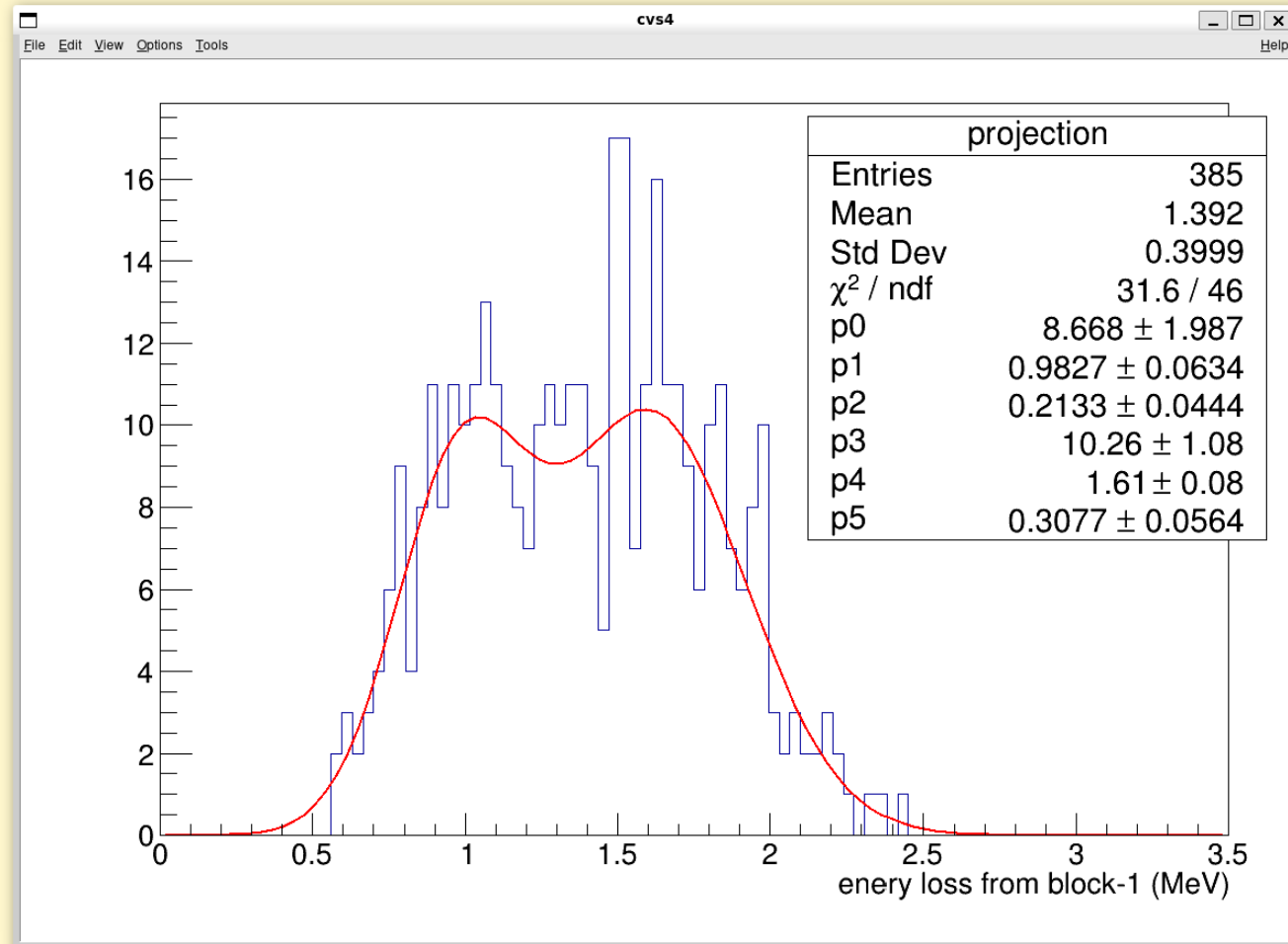
// 를 줄 앞에 붙여서 comment out

# root analysis\_3\_ee.C

- $^4\text{He}$  와  $^3\text{He}$  가 구분이 되는가?



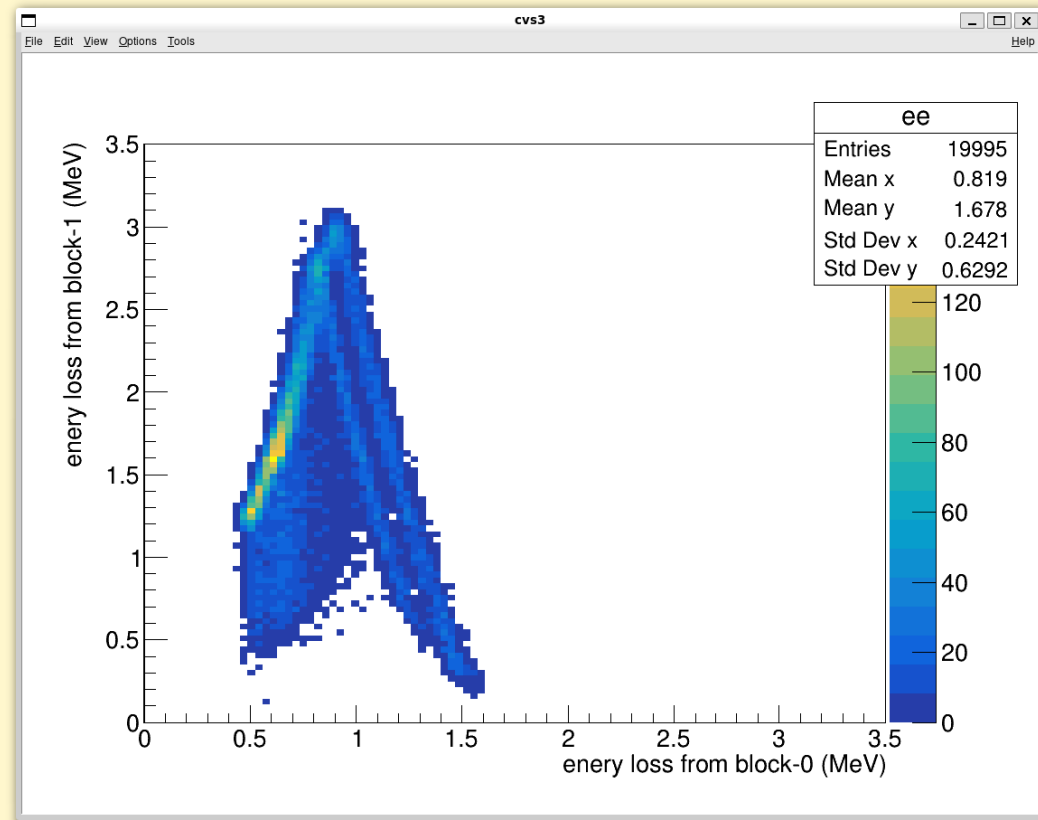
# root analysis\_4\_fit.C



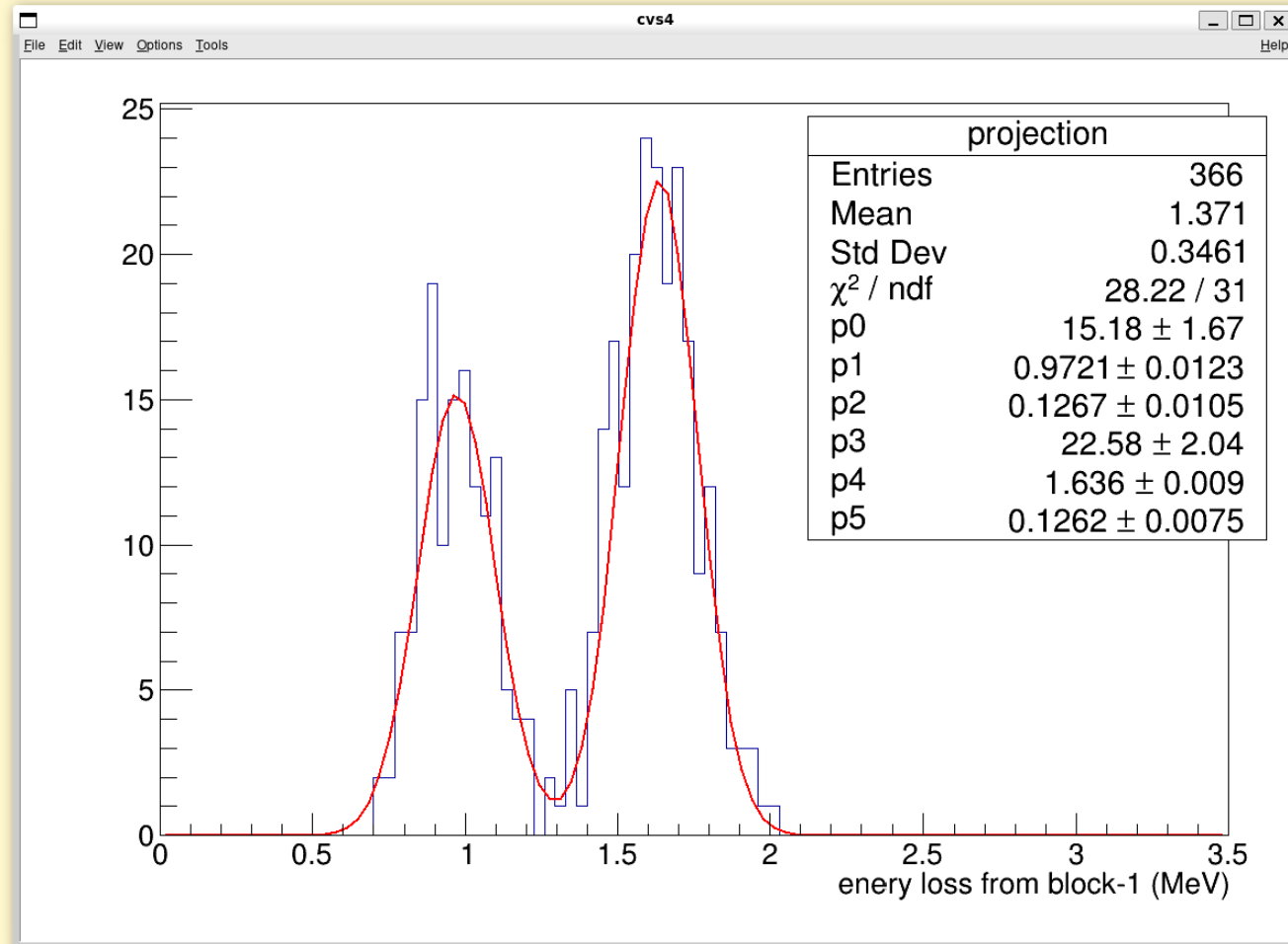


# root analysis\_3\_ee.C

- 지금까지 배운 실험의 설계에서 무엇을 개선한다면 다음과 같은 그림을 얻을 수 있을 지 생각해 보자.



# root analysis\_4\_fit.C



토론



# 토론

1. 시뮬레이션을 수월하게 수행하기 위해서 무엇을 최소화 하였는가?
2. 어떻게 시뮬레이션을 개선할 수 있는가?



끝!

