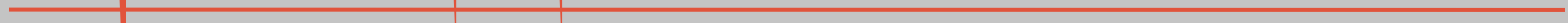


CA D AC F FC
FB AB

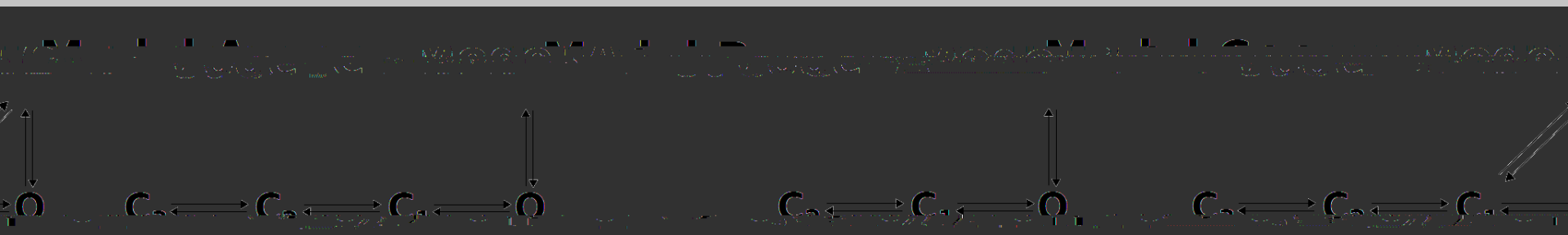


A
□ C
C
:C B , C D ,
D



D

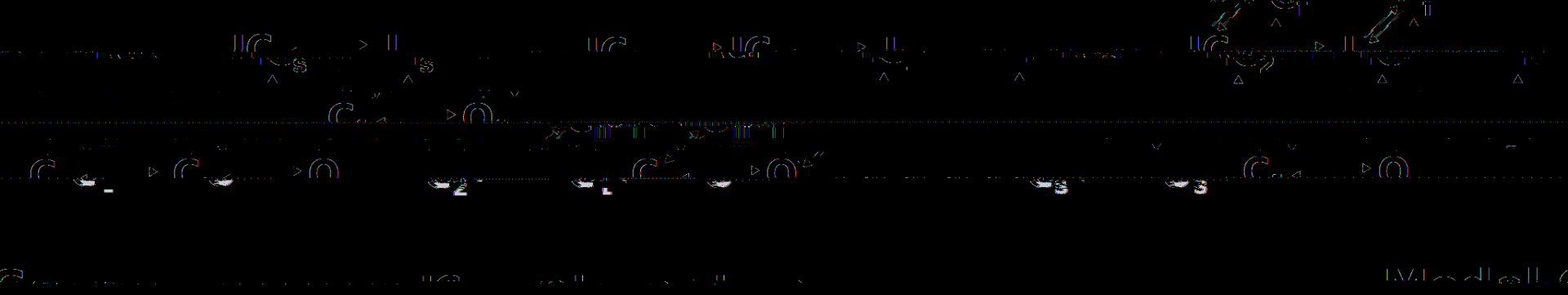
()



Model F

Model E

Model D



$$C =$$

$$=$$

$$=$$

D

,



(

DA ?)

A

B

A

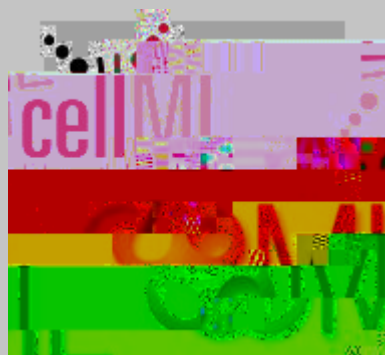
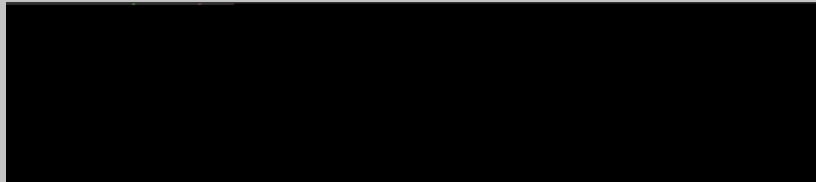
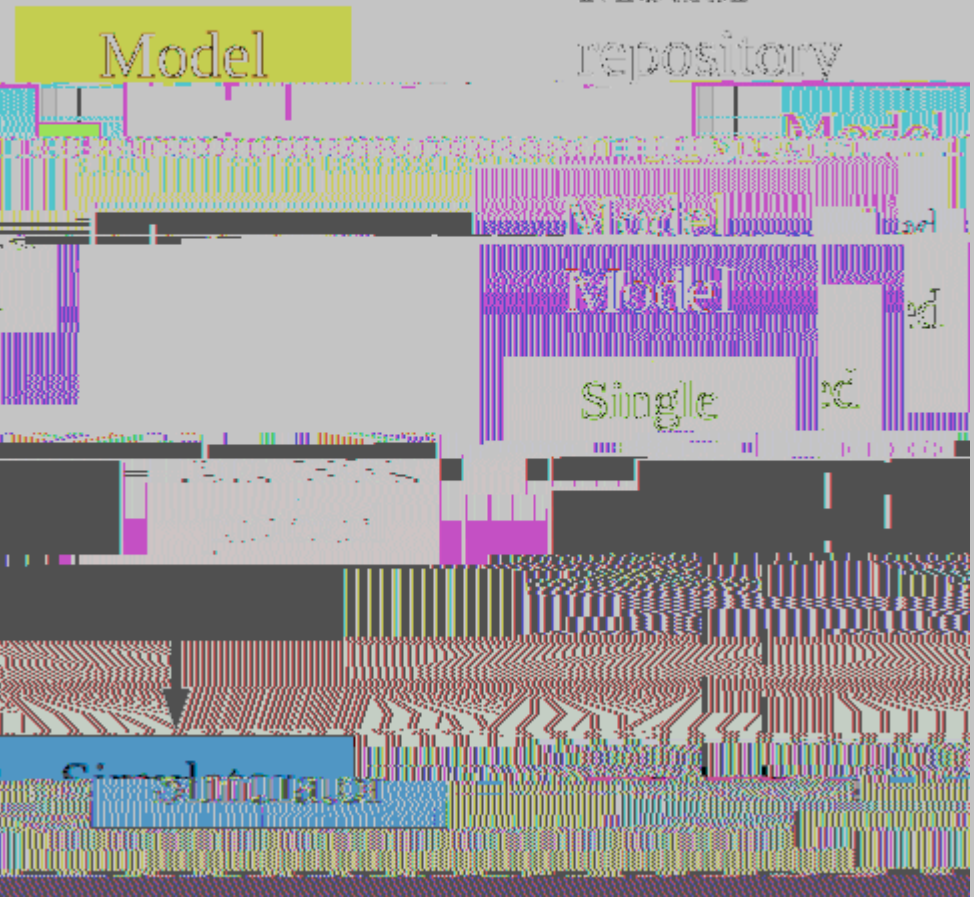
D

!



Model repository

Model repository



C

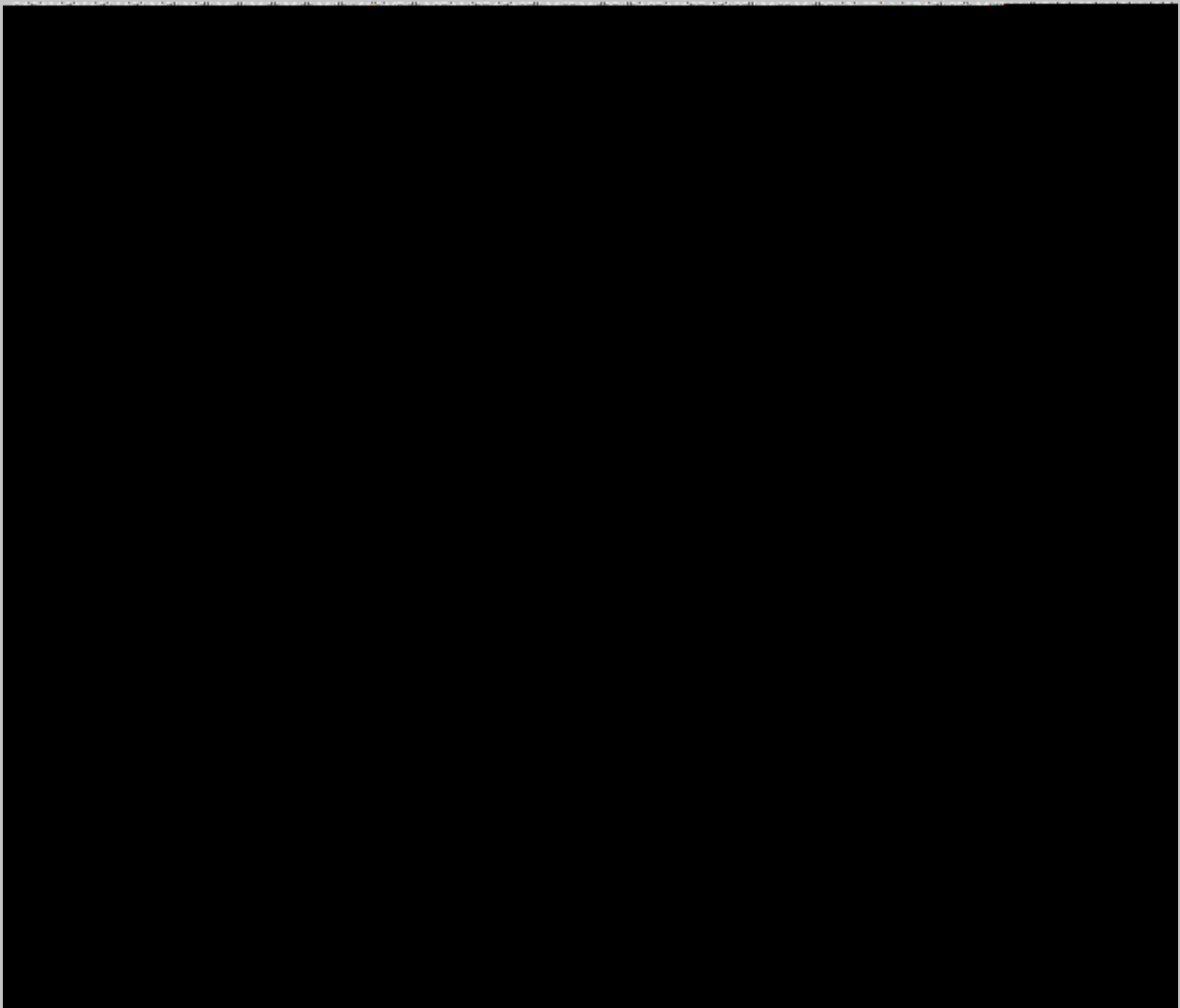
()

,

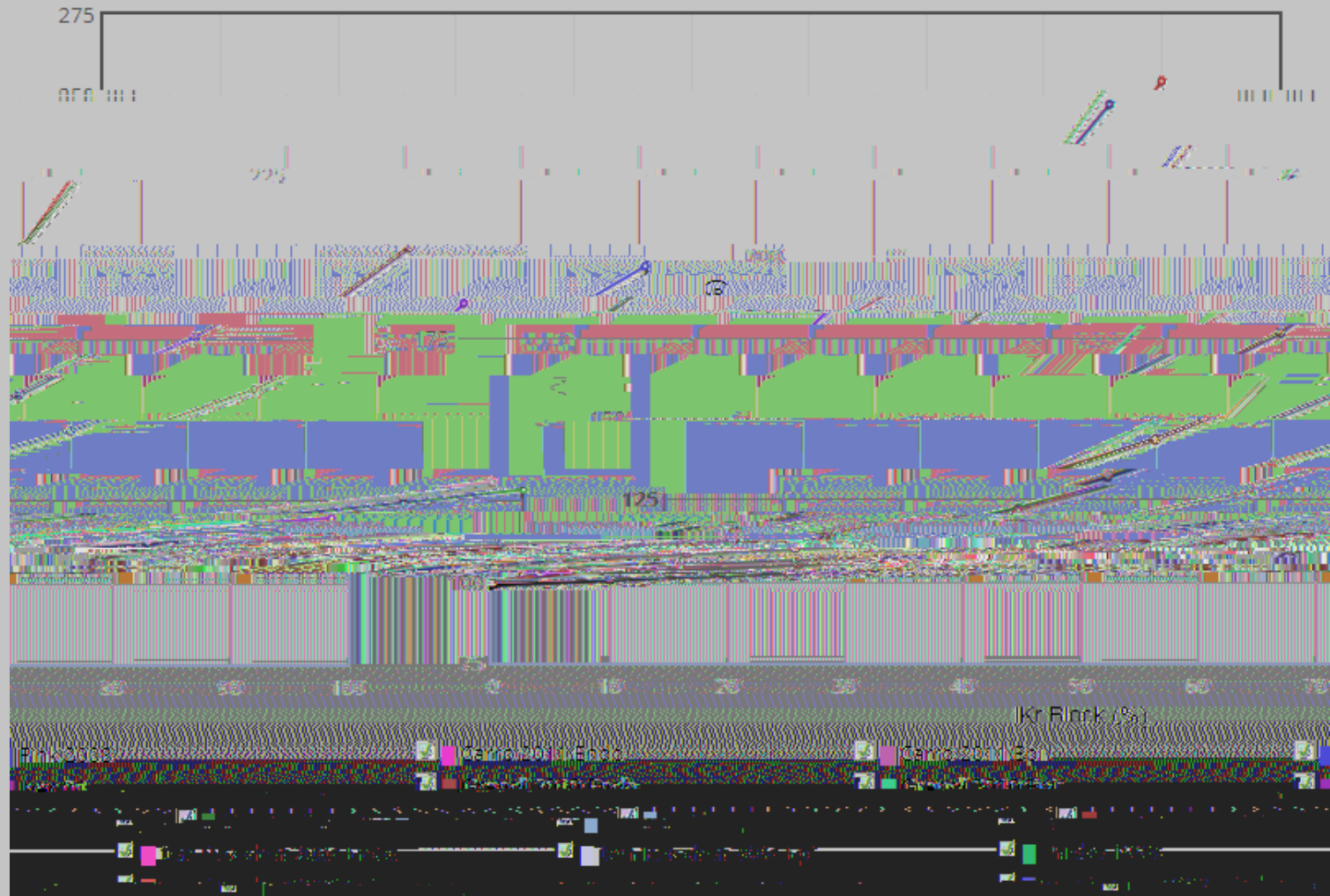
-

D

:// . . . /



C



Some

C & C A

, A , D

C & C DF

C++

&

C &

□ &

,

C

:

DF

C

F

-

DF

C

C

```

.      :
cdef extern from "nvector/nvector_serial.h":
    cdef N_Vector N_VMake_Serial (long int vec_length,
                                  real type *v_data)
    cdef struct _N_VectorContent_Serial:
        long int length
        real type *data
    ctypedef _N_VectorContent_Serial *N_VectorContent_Serial

cdef extern from "cvode/cvode.h":
    int CV_ADAMS

    ctypedef int (*CVRhsFn)(real type t, N_Vector y,
                            N_Vector ydot, void *user_data)
    void *CVodeCreate(int lmm, int iter)
    int CVode(void *cvode_mem, real type tout, N_Vector yout,
              real type *tret, int itask)

```

AC

DF

:. .

```
import numpy as np
```

```
class CcodeSolver:
```

```
    def __init__(self, cvode_solver_obj):
```

```
        self.cvode_solver_obj = cvode_solver_obj
```



AC

DF

::

```
import numpy as np
import numpy as np
import fc.sundials.sundials as _lib
```

```
cdef extern from "Python.h":
    object PyBuffer_FromReadWriteMemory(void *ptr, Py_ssize_t size)
```

```
cdef object NumpyView(N_Vector v):
    """Create a Numpy array giving a view on the CVODE vector passed in."""
    cdef _lib.N_VectorContent_Serial v_content =
        <_lib.N_VectorContent_Serial>(v.content)
    ret = np.empty(v_content.length, dtype=np.dtype)
    ret.data = PyBuffer_FromReadWriteMemory(v_content.data, ret.nbytes)
    return ret
```

```
self._state = _lib.N_VMake_Serial(self._state_size,
                                  <real type*>(np.ndarray)self.state).data)
flag = _lib.CVodeInit(self.cvode_mem, _RhsWrapper, 0.0, self._state)
```


:-

C++

"3* +4* ") ,

(

-
C

	IC L D	1 2 D
C++	197 (95)	201 (35)
	792	279
C	614 (583)	117 (54)
C	152 (125)	118 (27)
C++	266 (162)	204 (36)

■
■

□

—

C

C

*

/

C

A & ; &
C - -
A - ()
A

C

. :

```
def ScheduleExperiment(callbackUrl, signature, modelUrl, protoUrl,
                        user='', isAdmin=False):
    """Schedule a new experiment for execution."""
    from .tasks import CheckExperiment
    # Submit the job
    result = CheckExperiment.apply_async(
        (
```

⋮

A

□

*

&

B

!

C

D

□

C

(

)

,

&

&

⋮

C

-

,

-

,

&

,

!

A

,

&

D &



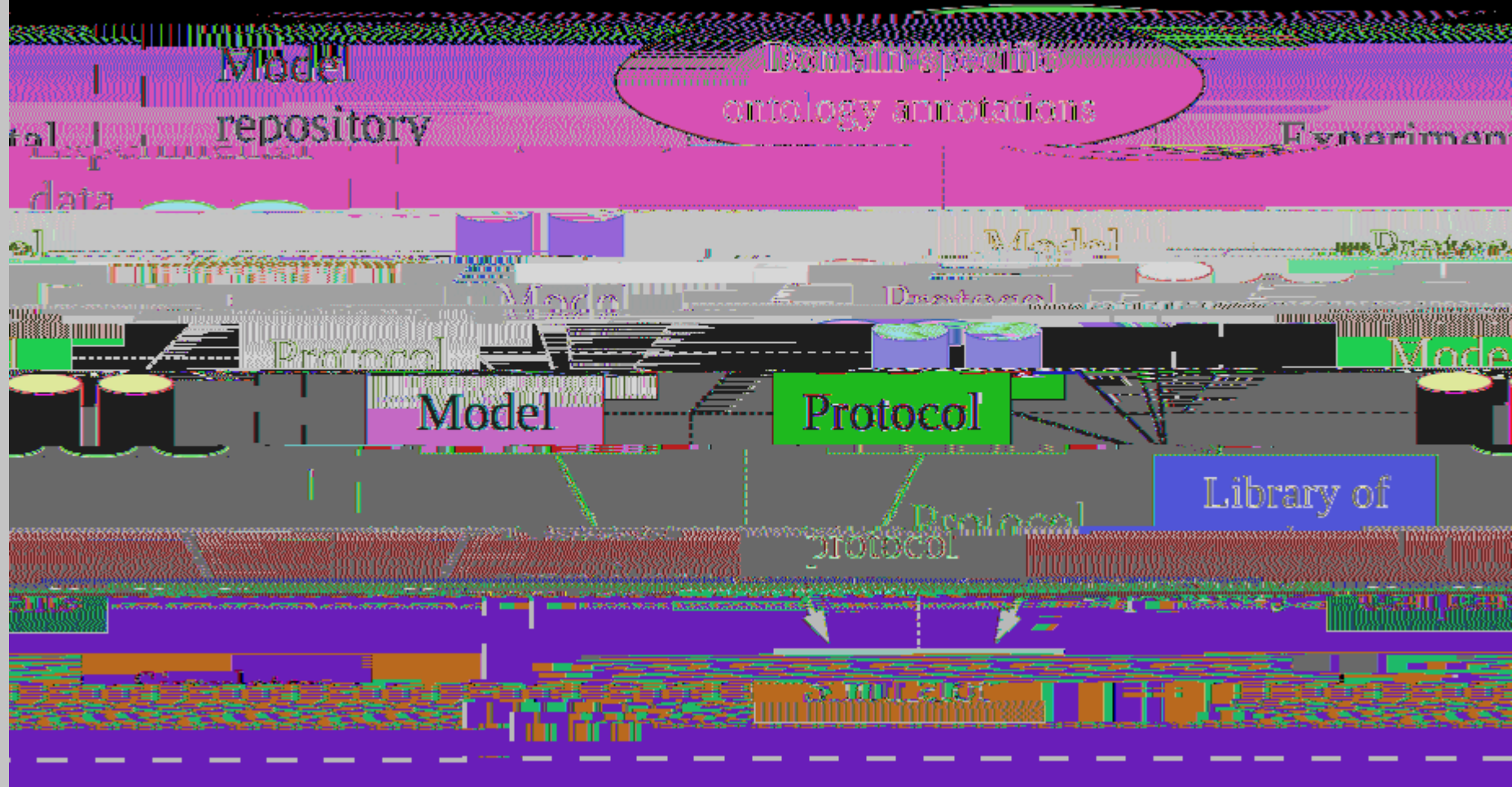
C

C

-

D 5,

Virtual laboratories with reusable virtual experiments



*



B

!

,

