Progress on Activity Analysis

Maksym Andriichuk, Petro Zarytskyi, Vassil Vassilev

Motivation

```
caas(x, y, z):
  a = x*y
  b = a*x
  return a
```

Do we need 'em?

```
caas_darg0(x, y, z):
  d_x = 1
  d_y = 0
  d_z = 0

  d_a = d_x*y + x*d_y
  a = x*y

  d_b = d_a*x + a*d_x
  b = a*x

  return d_a
```

Varied Analysis

Control Flow Graph

```
double cfg(double x, double y){
  double a = x*y;
  if(x==0 && y==0)
    return 0;
  else
    return a;
}
```

```
[B5 (ENTRY)]
  Succs (1): B4


[B4]
  1: double a = x * y;
  2: x == 0
  T: [B4.2] && ...
  Preds (1): B5
  Succs (2): B3 B1


[B3]
  1: y == 0
  T: if [B4.2] && [B3.1]
  Preds (1): B4
  Succs (2): B2 B1


[B2]
  1: 0 (ImplicitCastExpr, IntegralToFloating, double)
  2: return [B2.1];
  Preds (1): B3
  Succs (1): B0


[B1]
  1: a (ImplicitCastExpr, LValueToRValue, double)
  2: return [B1.1];
  Preds (2): B3 B4
  Succs (1): B0


[B0 (EXIT)]
  Preds (2): B1 B2
```

A variable is called ***varied*** if it depends on some independent input.

If a variable isn’t varied in the reverse mode the adjoint could be omitted.

```
double caas(double x){
  double a = x*x;
  double b = 1;
  b = b*b;
  return a;
}
```

```
void caas_grad(double x, double *_d_x) {
    double _d_a = 0.;
    double a = x * x;
    double _d_b = 0.;
    double b = 1;
    double _t0 = b;
    b = b * b;
    _d_a += 1;
    {
        b = _t0;
        double _r_d0 = _d_b;
        _d_b = 0.;
        _d_b += _r_d0 * b;
        _d_b += b * _r_d0;
    }
    {
        *_d_x += _d_a * x;
        *_d_x += x * _d_a;
    }
}
```

What is already done?

- Support for non-reference, non-pointer and non-array types of variables

```
double* ptr = &var;
```

Both are set varied

What is already done?

- Support for non-pointer and non-array types of variables
- Conditional statements
- Conditions

```
double caas(double x){
  double a, b = 0;
  if(x)
    a = x;
  else
    b = a;
}
```

What is already done?

- Support for non-pointer and non-array types of variables
- Conditional statements
- Conditions
- Loops

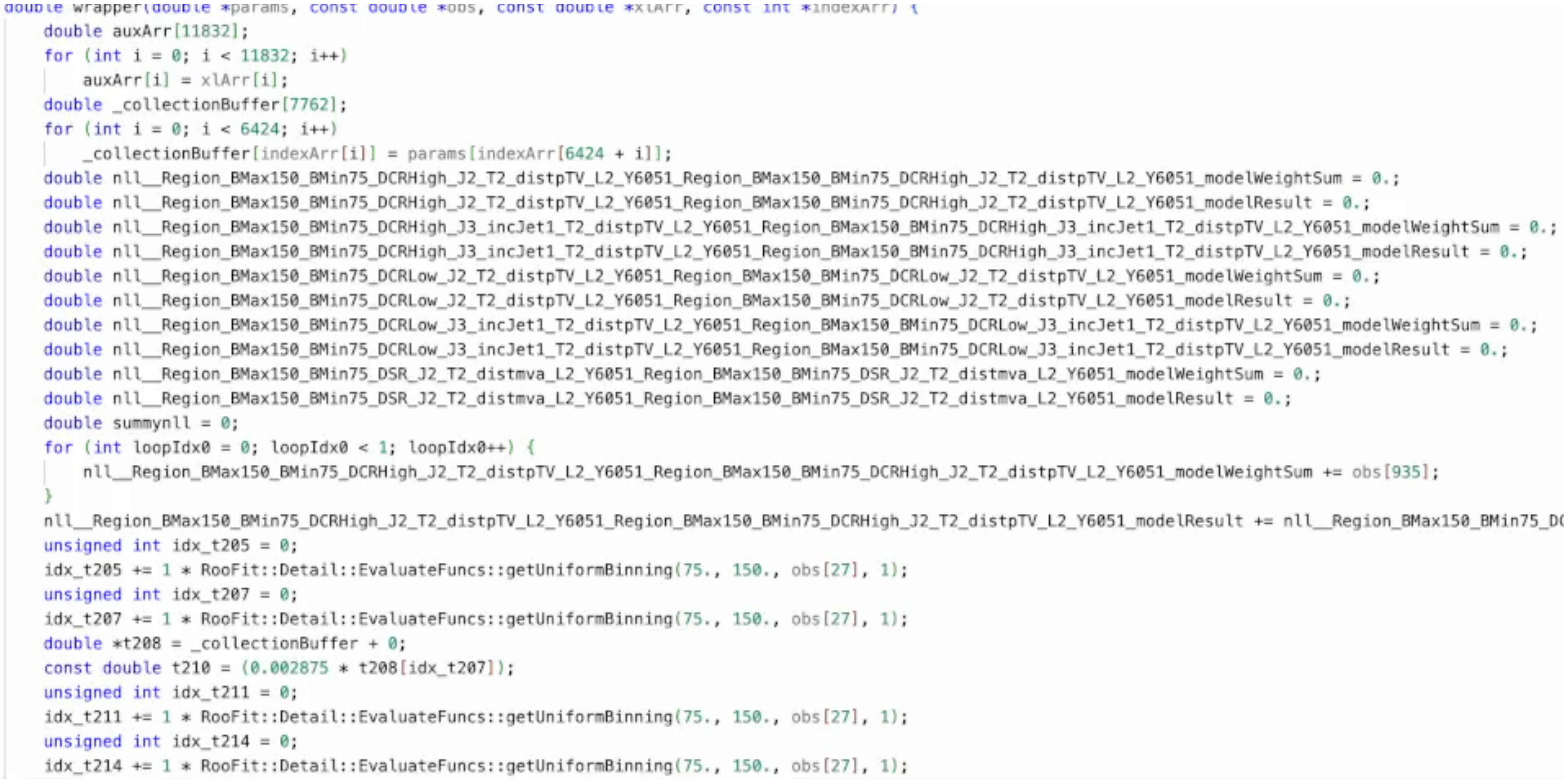
```
double f3(double x){
  double x1, x2, x3, x4, x5 = 0;
  while(!x3){
    x5 = x4;
    x4 = x3;
    x3 = x2;
    x2 = x1;
    x1 = x;
  }
  return x5;
}
```

What is already done?

- Support for non-pointer and non-array types of variables
- Conditional statements
- Conditions
- Loops
- Primitive function calls support

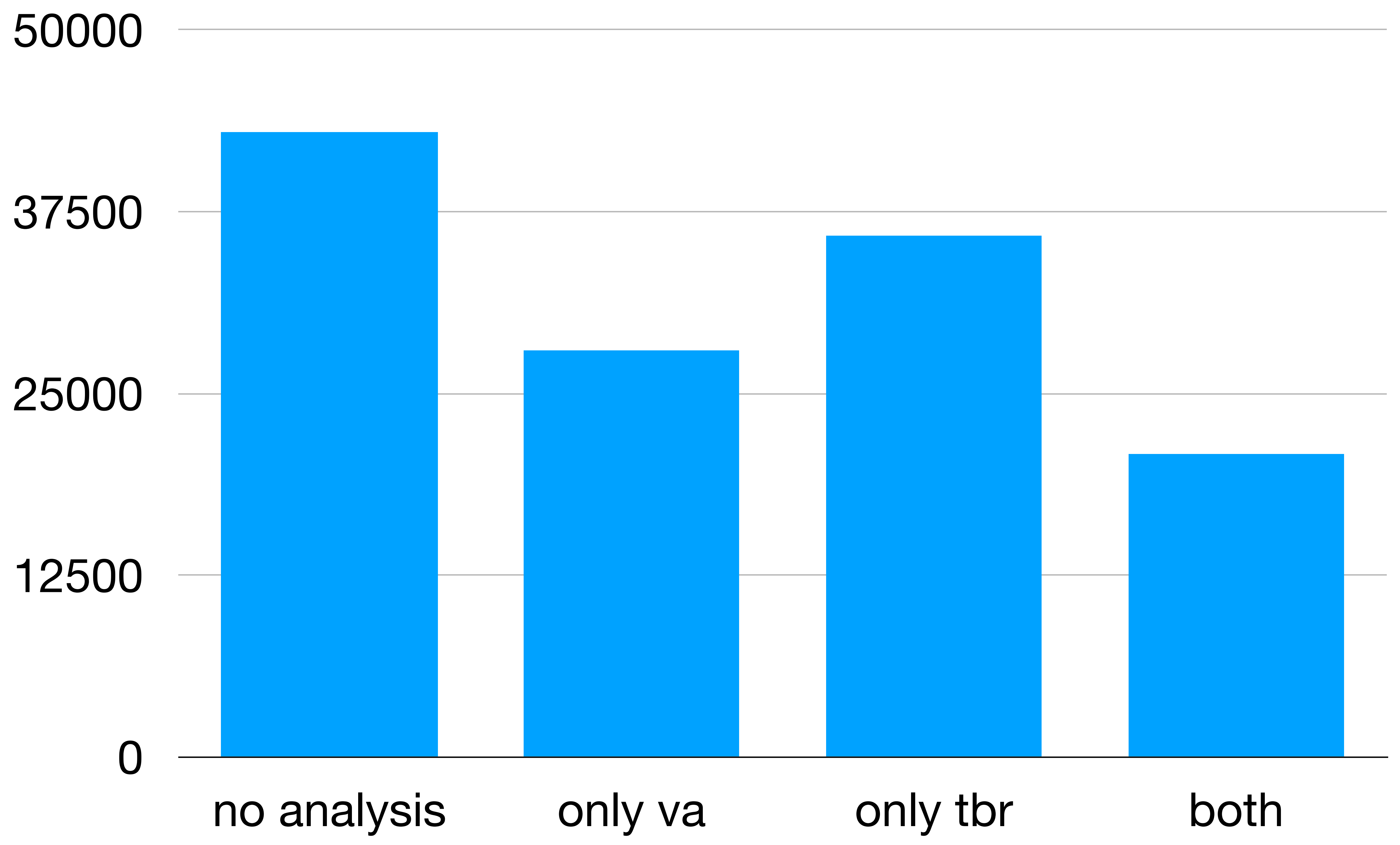
Challenges

- Objects and member functions
- References
- Function calls
- Lambda functions and functors support
- Pointers
- Useful Analysis



```
double wrapper(double *params, const double *obs, const double *xlArr, const int *indexArr) {
    double auxArr[11832];
    for (int i = 0; i < 11832; i++)
        auxArr[i] = xlArr[i];
    double _collectionBuffer[7762];
    for (int i = 0; i < 6424; i++)
        _collectionBuffer[indexArr[i]] = params[indexArr[6424 + i]];
    double nll__Region_BMax150_BMin75_DCRHigh_J2_T2_distpTV_L2_Y6051_Region_BMax150_BMin75_DCRHigh_J2_T2_distpTV_L2_Y6051_modelWeightSum = 0.;
    double nll__Region_BMax150_BMin75_DCRHigh_J2_T2_distpTV_L2_Y6051_Region_BMax150_BMin75_DCRHigh_J2_T2_distpTV_L2_Y6051_modelResult = 0.;
    double nll__Region_BMax150_BMin75_DCRHigh_J3_incJet1_T2_distpTV_L2_Y6051_Region_BMax150_BMin75_DCRHigh_J3_incJet1_T2_distpTV_L2_Y6051_modelWeightSum = 0.;
    double nll__Region_BMax150_BMin75_DCRHigh_J3_incJet1_T2_distpTV_L2_Y6051_Region_BMax150_BMin75_DCRHigh_J3_incJet1_T2_distpTV_L2_Y6051_modelResult = 0.;
    double nll__Region_BMax150_BMin75_DCRLow_J2_T2_distpTV_L2_Y6051_Region_BMax150_BMin75_DCRLow_J2_T2_distpTV_L2_Y6051_modelWeightSum = 0.;
    double nll__Region_BMax150_BMin75_DCRLow_J2_T2_distpTV_L2_Y6051_Region_BMax150_BMin75_DCRLow_J2_T2_distpTV_L2_Y6051_modelResult = 0.;
    double nll__Region_BMax150_BMin75_DCRLow_J3_incJet1_T2_distpTV_L2_Y6051_Region_BMax150_BMin75_DCRLow_J3_incJet1_T2_distpTV_L2_Y6051_modelWeightSum = 0.;
    double nll__Region_BMax150_BMin75_DCRLow_J3_incJet1_T2_distpTV_L2_Y6051_Region_BMax150_BMin75_DCRLow_J3_incJet1_T2_distpTV_L2_Y6051_modelResult = 0.;
    double nll__Region_BMax150_BMin75_DSR_J2_T2_distmva_L2_Y6051_Region_BMax150_BMin75_DSR_J2_T2_distmva_L2_Y6051_modelWeightSum = 0.;
    double nll__Region_BMax150_BMin75_DSR_J2_T2_distmva_L2_Y6051_Region_BMax150_BMin75_DSR_J2_T2_distmva_L2_Y6051_modelResult = 0.;
    double summynll = 0;
    for (int loopIdx0 = 0; loopIdx0 < 1; loopIdx0++) {
        nll__Region_BMax150_BMin75_DCRHigh_J2_T2_distpTV_L2_Y6051_Region_BMax150_BMin75_DCRHigh_J2_T2_distpTV_L2_Y6051_modelWeightSum += obs[935];
    }
    nll__Region_BMax150_BMin75_DCRHigh_J2_T2_distpTV_L2_Y6051_Region_BMax150_BMin75_DCRHigh_J2_T2_distpTV_L2_Y6051_modelResult += nll__Region_BMax150_BMin75_D
    unsigned int idx_t205 = 0;
    idx_t205 += 1 * RooFit::Detail::EvaluateFuncs::getUniformBinning(75., 150., obs[27], 1);
    unsigned int idx_t207 = 0;
    idx_t207 += 1 * RooFit::Detail::EvaluateFuncs::getUniformBinning(75., 150., obs[27], 1);
    double *t208 = _collectionBuffer + 0;
    const double t210 = (0.002875 * t208[idx_t207]);
    unsigned int idx_t211 = 0;
    idx_t211 += 1 * RooFit::Detail::EvaluateFuncs::getUniformBinning(75., 150., obs[27], 1);
    unsigned int idx_t214 = 0;
    idx_t214 += 1 * RooFit::Detail::EvaluateFuncs::getUniformBinning(75., 150., obs[27], 1);
```

Preliminary Results


no analysis
only va
only tbr
both