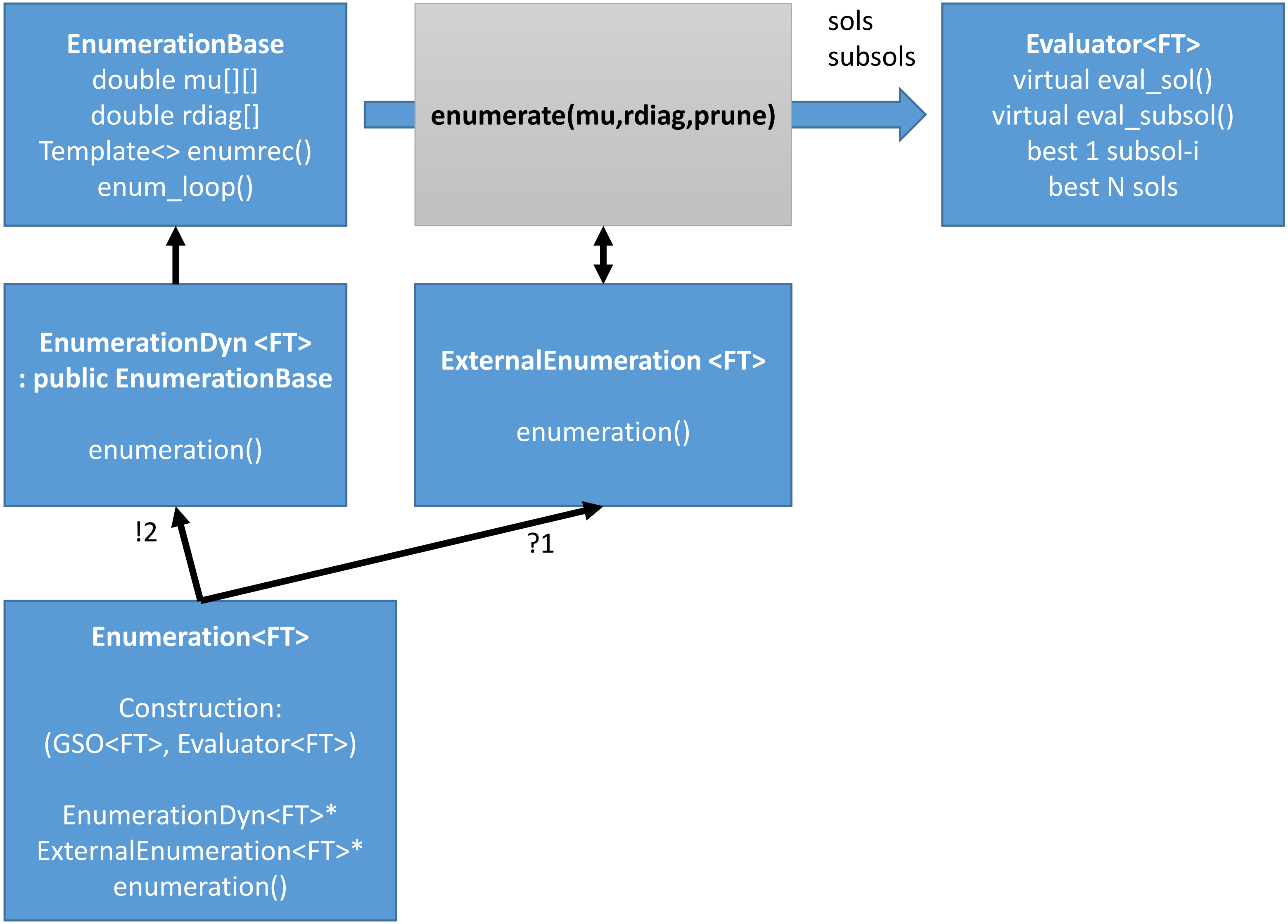


EnumerationBase
double mu[][]
double rdiag[]
Template<> enumrec()
enum_loop()

enumerate(mu,rdiag,prune)

sols
subsols

Evaluator<FT>
virtual eval_sol()
virtual eval_subsol()
best 1 subsol-i
best N sols

EnumerationDyn <FT>
: public EnumerationBase

enumeration()

ExternalEnumeration <FT>

enumeration()

!2

?1

Enumeration<FT>

Construction:
(GSO<FT>, Evaluator<FT>)

EnumerationDyn<FT>*
ExternalEnumeration<FT>*
enumeration()

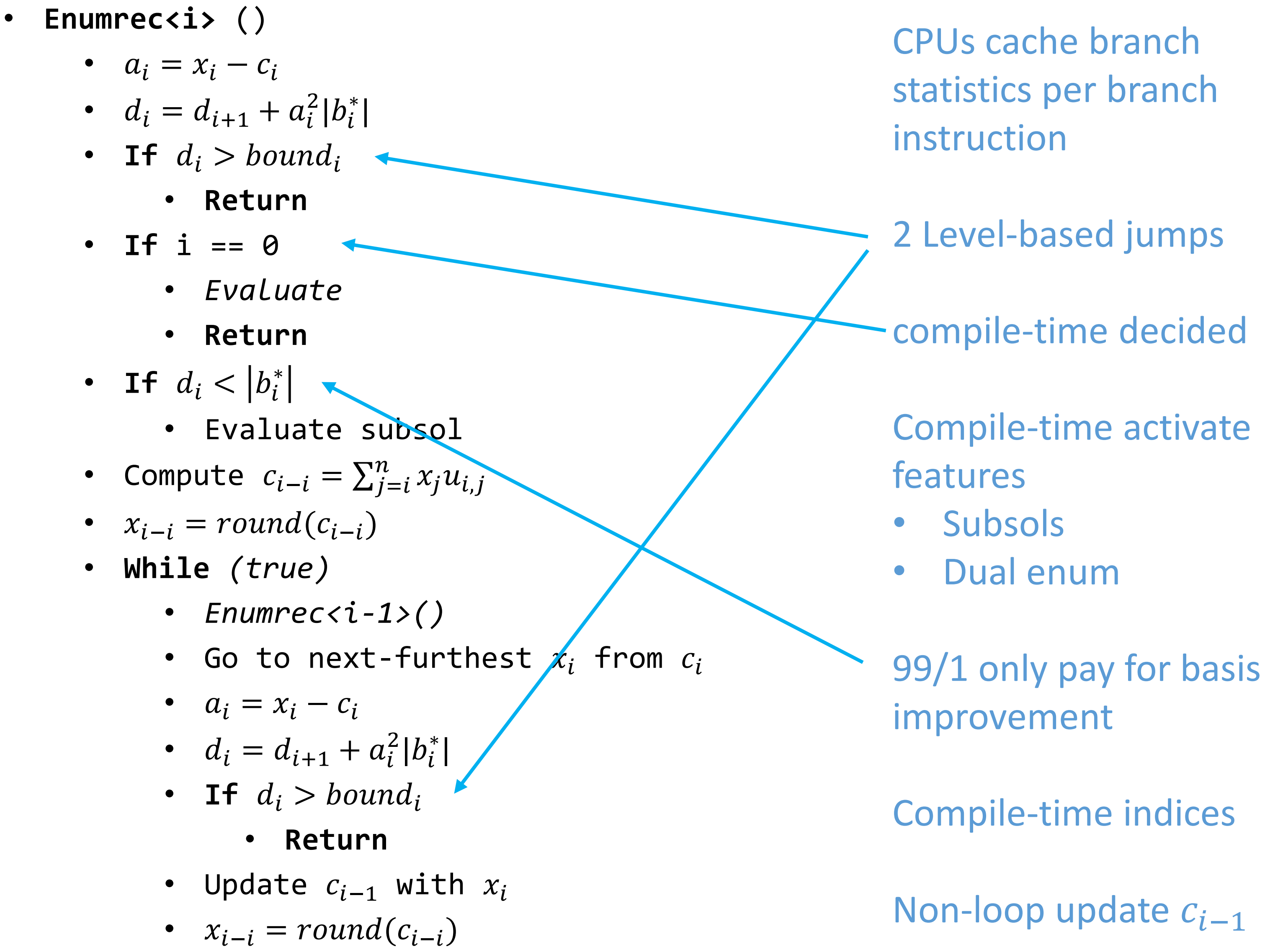
```
• While ()
    • $a_i = x_i - c_i$
    • $d_i = d_{i+1} + a_i^2 |b_i^*|$
    • If $d_i < bound_i$
        • --i
        • If i < 0:
            • Evaluate
            • Backtrack
        • Compute $c_i = \sum_{j=i+1}^{n} x_j u_{i,j}$
        • $x_i = round(c_i)$
    • Else
        • Backtrack

• Backtrack
    • ++i
    • If i==n: exit
    • Go to next-furthest $x_i$ from $c_i$
```

CPUs cache branch statistics per branch instruction

50/50 global averaged

0/100



```
- Enumrec<i> ()
    - $a_i = x_i - c_i$
    - $d_i = d_{i+1} + a_i^2 |b_i^*|$
    - If $d_i > bound_i$
        - Return
    - If i == 0
        - Evaluate
        - Return
    - If $d_i < |b_i^*|$
        - Evaluate subsol
    - Compute $c_{i-i} = \sum_{j=i}^{n} x_j u_{i,j}$
    - $x_{i-i} = round(c_{i-i})$
    - While (true)
        - Enumrec<i-1>()
        - Go to next-furthest $x_i$ from $c_i$
        - $a_i = x_i - c_i$
        - $d_i = d_{i+1} + a_i^2 |b_i^*|$
        - If $d_i > bound_i$
            - Return
        - Update $c_{i-1}$ with $x_i$
        - $x_{i-i} = round(c_{i-i})$
```

CPUs cache branch statistics per branch instruction

2 Level-based jumps

compile-time decided

Compile-time activate features

- Subsols
- Dual enum

99/1 only pay for basis improvement

Compile-time indices

Non-loop update c_{i-1}

External Enumeration function API

```
typedef uint64_t(extenum_fc_enumerate)(int dim, enumf maxdist,
                                       std::function<extenum_cb_set_config> cbfunc,
                                       std::function<extenum_cb_process_sol> cbsol,
                                       std::function<extenum_cb_process_subsol> cbsubsol,
                                       bool dual /*=false*/, bool findsubsols /*=false*/
                                       );
// return 0 for failed/not-supported or #nodes>0
void set_external_enumerator(std::function<extenum_fc_enumerate> extenum = nullptr);
```

ExternalEnumeration provided callback functions for external enumeration functions:

```
typedef void(extenum_cb_set_config)(enumf *mu, size_t mudim, bool mutranspose, enumf *rdiag,
                                    enumf *pruning);
```

```
typedef enumf(extenum_cb_process_sol)(enumf dist, enumf *sol);
```

```
typedef void(extenum_cb_process_subsol)(enumf dist, enumf *subsol, int offset);
```