ZX-Calculus operations external to full_reduce method:
bialg_simp | clifford_simp | gadget_simp | id_simp | lcomp_simp | phase_free_simp |
pivot_boundary_simp | pivot_gadget_simp | pivot_simp | spider_simp | supplementarity_simp

Full_reduce method (fixed order loops):
interior_clifford_simp(g, matchf=matchf, quiet=quiet, stats=stats)
pivot_gadget_simp(g, matchf=matchf, quiet=quiet, stats=stats)
while True:
    clifford_simp(g, matchf=matchf, quiet=quiet, stats=stats)
    i = **gadget_simp**(g, matchf=matchf, quiet=quiet, stats=stats)
    interior_clifford_simp(g, matchf=matchf, quiet=quiet, stats=stats)
    j = **pivot_gadget_simp**(g, matchf=matchf, quiet=quiet, stats=stats)
    if i+j == 0:
        Break

Interior_clifford_simp method (fixed order loops):
spider_simp(g, matchf=matchf, quiet=quiet, stats=stats)
to_gh(g)
i = 0
while True:
    i1 = **id_simp**(g, matchf=matchf, quiet=quiet, stats=stats)
    i2 = **spider_simp**(g, matchf=matchf, quiet=quiet, stats=stats)
    i3 = **pivot_simp**(g, matchf=matchf, quiet=quiet, stats=stats)
    i4 = **lcomp_simp**(g, matchf=matchf, quiet=quiet, stats=stats)
    if i1+i2+i3+i4==0: break
    i += 1
  return i

Pivot_gadget_simp method (manipulating graph properties):
def pivot_gadget_simp(g: BaseGraph[VT,ET], matchf:Optional[Callable[[ET],bool]]=None,
quiet:bool=True, stats:Optional[Stats]=None) -> int:
  return simp(g, 'pivot_gadget_simp', match_pivot_gadget, pivot,
          auto_simplify_parallel_edges=True, matchf=matchf, quiet=quiet, stats=stats)

Clifford_simp method (fixed order loops):
i = 0
while True:
    i += **interior_clifford_simp**(g, matchf=matchf, quiet=quiet, stats=stats)
    i2 = **pivot_boundary_simp**(g, matchf=matchf, quiet=quiet, stats=stats)
    if i2 == 0:
        break
  return i

Gadget_simp method (manipulating graph properties):

```
def gadget_simp(g: BaseGraph[VT,ET], matchf: Optional[Callable[[VT],bool]]=None,
quiet:bool=True, stats:Optional[Stats]=None) -> int:
    return simp(g, 'gadget_simp', match_phase_gadgets, merge_phase_gadgets,
            auto_simplify_parallel_edges=True, matchf=matchf, quiet=quiet, stats=stats)
```

Pivot_boundary_simp (manipulating graph properties):

```
def pivot_boundary_simp(g: BaseGraph[VT,ET], matchf:Optional[Callable[[ET],bool]]=None,
quiet:bool=True, stats:Optional[Stats]=None) -> int:
    return simp(g, 'pivot_boundary_simp', match_pivot_boundary, pivot,
            auto_simplify_parallel_edges=True, matchf=matchf, quiet=quiet, stats=stats)
```