```
QZip_{ℰs_List}@𝔼[L_, Q_, P_] := Module[{ξ, z, zs, c, ys, ηs, qt, zrule, ℰrule, out},
    zs = Table[ξ*, {ξ, ℰs}];
    c = CF[Q /. Alternatives @@ (ℰs ⋃ zs) → 0];
    ys = CF@Table[∂_ξ (Q /. Alternatives @@ zs → 0), {ξ, ℰs}];
    ηs = CF@Table[∂_z (Q /. Alternatives @@ ℰs → 0), {z, zs}];
    qt = CF@Inverse@Table[Kδ_{z,ξ*} − ∂_{z,ξ}Q, {ξ, ℰs}, {z, zs}];
    zrule = Thread[zs → CF[qt.(zs + ys)]];
    ℰrule = Thread[ℰs → ℰs + ηs.qt];
    CF /@ 𝔼[L, c + ηs.qt.ys, Det[qt] Zip_{ℰs}[P /. (zrule ⋃ ℰrule)]] ];
```