The Casimir trick simplifies spinor products that are summed over polarizations.

$$\sum_{s_1=1}^{2}\sum_{s_2=1}^{2}\sum_{s_3=1}^{2}\sum_{s_4=1}^{2}(\bar{v}_4\gamma_\mu u_3)(\bar{u}_1\gamma^\mu v_2)(\bar{u}_3\gamma_\nu v_4)(\bar{v}_2\gamma^\nu u_1)$$
$$= N\,\mathrm{Tr}\left[(\not{p}_1+m_1)\gamma^\mu(\not{p}_2-m_2)\gamma^\nu\right]\mathrm{Tr}\left[(\not{p}_4-m_4)\gamma_\mu(\not{p}_3+m_3)\gamma_\nu\right]$$

The index s_1 is the spin of u_1, s_2 of v_2, etc. The above equation has a unified normalization constant N on the right-hand side instead of normalizing each spinor by a factor of $1/\sqrt{E+m}$. This is because the software is better at simplifying expressions that have multipliers instead of divisors. There are four spinors hence the normalization constant is

$$N = (E_1+m_1)(E_2+m_2)(E_3+m_3)(E_4+m_4)$$

The script below proves the Casimir trick by direct calculation. The following notes clarify what the script is doing.

(1) Spinors u_1 and u_3 are fermions, v_2 and v_4 are anti-fermions. For summing over spins, index `[1]` selects spin up, `[2]` selects spin down.

$$\texttt{u1[1]} = \begin{pmatrix} E_1+m_1 \\ 0 \\ p_{1z} \\ p_{1x}+ip_{iy} \end{pmatrix} \quad \texttt{v2[1]} = \begin{pmatrix} p_{2z} \\ p_{2x}+ip_{2y} \\ E_2+m_2 \\ 0 \end{pmatrix} \quad \texttt{u3[1]} = \begin{pmatrix} E_3+m_3 \\ 0 \\ p_{3z} \\ p_{3x}+ip_{3y} \end{pmatrix} \quad \texttt{v4[1]} = \begin{pmatrix} p_{4z} \\ p_{4x}+ip_{4y} \\ E_4+m_4 \\ 0 \end{pmatrix}$$

$$\texttt{u1[2]} = \begin{pmatrix} 0 \\ E_1+m_1 \\ p_{1x}-ip_{1y} \\ -p_{1z} \end{pmatrix} \quad \texttt{v2[2]} = \begin{pmatrix} p_{2x}-ip_{2y} \\ -p_{2z} \\ 0 \\ E_2+m_2 \end{pmatrix} \quad \texttt{u3[2]} = \begin{pmatrix} 0 \\ E_3+m_3 \\ p_{3x}-ip_{3y} \\ -p_{3z} \end{pmatrix} \quad \texttt{v4[2]} = \begin{pmatrix} p_{4x}-ip_{4y} \\ -p_{4z} \\ 0 \\ E_4+m_4 \end{pmatrix}$$

(2) Recall that spinors u and v are not spacetime vectors and are not associated with the spacetime metric $g_{\mu\nu}$. However γ^μ is a spacetime tensor and requires $g_{\mu\nu}$ to lower the index. In component notation we have

$$(\bar{v}_4\gamma_\mu u_3)(\bar{u}_1\gamma^\mu v_2) = g_{\mu\nu}\left(\bar{v}_{4\alpha}\gamma^{\nu\alpha}{}_\beta u_3{}^\beta\right)\left(\bar{u}_{1\rho}\gamma^{\mu\rho}{}_\sigma v_2{}^\sigma\right)$$

Transpose γ to form inner products.

$$= g_{\mu\nu}\left(\bar{v}_{4\alpha}\gamma^{\alpha\nu}{}_\beta u_3{}^\beta\right)\left(\bar{u}_{1\rho}\gamma^{\rho\mu}{}_\sigma v_2{}^\sigma\right)$$

Convert to code.

$$= \underset{g_{\mu\nu}}{\texttt{dot(gmunu}}\texttt{,dot(}\underbrace{\texttt{bar(v4[s4])}}_{\bar{v}_4}\texttt{,}\underset{\gamma^{\alpha\nu}{}_\beta}{\texttt{gammaT}}\texttt{,}\underbrace{\texttt{u3[s3]}}_{u_3}\texttt{),dot(}\underbrace{\texttt{bar(u1[s1])}}_{\bar{u}_1}\texttt{,}\underset{\gamma^{\rho\mu}{}_\sigma}{\texttt{gammaT}}\texttt{,}\underbrace{\texttt{v2[s2]}}_{v_2}\texttt{))}$$

Repeat for the remaining spinor products.

$$(\bar{u}_3\gamma_\nu v_4)(\bar{v}_2\gamma^\nu u_1)$$
$$= \texttt{dot(gmunu,dot(bar(u3[s3]),gammaT,v4[s4]),dot(bar(v2[s2]),gammaT,u1[s1]))}$$

(3) The traces are 4×4 matrices. In component notation we have

$$\operatorname{Tr}\left[(\not p_1 + m_1)\gamma^\mu(\not p_2 - m_2)\gamma^\nu\right] = (\not p_1 + m_1)^\alpha{}_\beta \gamma^{\mu\beta}{}_\rho (\not p_2 - m_2)^\rho{}_\sigma \gamma^{\nu\sigma}{}_\alpha$$

Transpose γ to form inner products.

$$= (\not p_1 + m_1)^\alpha{}_\beta \gamma^{\beta\mu}{}_\rho (\not p_2 - m_2)^\rho{}_\sigma \gamma^{\sigma\nu}{}_\alpha$$

Convert to code. Contract sums over α.

$$= \texttt{contract(dot(}\underbrace{\texttt{pslash1+m1*I}}_{(\not p_1+m_1)^\alpha{}_\beta}\texttt{,}\underbrace{\texttt{gammaT}}_{\gamma^{\beta\mu}{}_\rho}\texttt{,}\underbrace{\texttt{pslash2-m2*I}}_{(\not p_2-m_2)^\rho{}_\sigma}\texttt{,}\underbrace{\texttt{gammaT}}_{\gamma^{\sigma\nu}{}_\alpha}\texttt{),1,4)}$$

Repeat for the second trace.

$$\operatorname{Tr}\left[(\not p_4 - m_4)\gamma^\mu(\not p_3 + m_3)\gamma^\nu\right]$$
$$= \texttt{contract(dot(pslash4-m4*I,gammaT,pslash3+m3*I,gammaT),1,4)}$$

(4) The product of the two traces is summed over μ and ν.

$$\operatorname{Tr}[\,]\operatorname{Tr}[\,] = \operatorname{Tr}[\,]^{\mu\nu}\operatorname{Tr}[\,]_{\mu\nu}$$

Transpose the second trace to form an inner product.

$$= \operatorname{Tr}[\,]^{\mu\nu}\operatorname{Tr}[\,]_{\nu\mu}$$

Convert to code. Contract sums over μ.

$$= \texttt{contract(dot(T1,transpose(T2)))}$$

```
-- www.eigenmath.org/casimir.txt
E1 = sqrt(p1x^2 + p1y^2 + p1z^2 + m1^2)
E2 = sqrt(p2x^2 + p2y^2 + p2z^2 + m2^2)
E3 = sqrt(p3x^2 + p3y^2 + p3z^2 + m3^2)
E4 = sqrt(p4x^2 + p4y^2 + p4z^2 + m4^2)
p1 = (E1,p1x,p1y,p1z)
p2 = (E2,p2x,p2y,p2z)
p3 = (E3,p3x,p3y,p3z)
p4 = (E4,p4x,p4y,p4z)
u11 = (E1+m1,0,p1z,p1x+i*p1y)
u12 = (0,E1+m1,p1x-i*p1y,-p1z)
v21 = (p2z,p2x+i*p2y,E2+m2,0)
v22 = (p2x-i*p2y,-p2z,0,E2+m2)
u31 = (E3+m3,0,p3z,p3x+i*p3y)
u32 = (0,E3+m3,p3x-i*p3y,-p3z)
v41 = (p4z,p4x+i*p4y,E4+m4,0)
v42 = (p4x-i*p4y,-p4z,0,E4+m4)
u1 = (u11,u12)
```

```
v2 = (v21,v22)
u3 = (u31,u32)
v4 = (v41,v42)
I = ((1,0,0,0),(0,1,0,0),(0,0,1,0),(0,0,0,1))
gmunu = ((1,0,0,0),(0,-1,0,0),(0,0,-1,0),(0,0,0,-1))
gamma0 = ((1,0,0,0),(0,1,0,0),(0,0,-1,0),(0,0,0,-1))
gamma1 = ((0,0,0,1),(0,0,1,0),(0,-1,0,0),(-1,0,0,0))
gamma2 = ((0,0,0,-i),(0,0,i,0),(0,i,0,0),(-i,0,0,0))
gamma3 = ((0,0,1,0),(0,0,0,-1),(-1,0,0,0),(0,1,0,0))
gamma = (gamma0,gamma1,gamma2,gamma3)
pslash1 = dot(gmunu,p1,gamma)
pslash2 = dot(gmunu,p2,gamma)
pslash3 = dot(gmunu,p3,gamma)
pslash4 = dot(gmunu,p4,gamma)
gammaT = transpose(gamma)
bar(u) = dot(conj(u),gamma0)
-- A is the sum over spins
A = sum(s1,1,2,sum(s2,1,2,sum(s3,1,2,sum(s4,1,2,
  dot(gmunu,dot(bar(v4[s4]),gammaT,u3[s3]),dot(bar(u1[s1]),gammaT,v2[s2])) *
  dot(gmunu,dot(bar(u3[s3]),gammaT,v4[s4]),dot(bar(v2[s2]),gammaT,u1[s1]))
))))
-- T1 is the first trace matrix
T1 = contract(dot(pslash1+m1*I,gammaT,pslash2-m2*I,gammaT),1,4)
-- T2 is the second trace matrix
T2 = contract(dot(pslash4-m4*I,gammaT,pslash3+m3*I,gammaT),1,4)
T2 = dot(gmunu,T2,gmunu)
-- B is the product of T1 and T2
B = contract(dot(T1,transpose(T2)))
-- N is the normalization constant
N = (E1+m1)*(E2+m2)*(E3+m3)*(E4+m4)
-- this difference should be zero
A-N*B
```