Color Dressed One-Loop Partonic Amplitudes

In collaboration with
Jan Winter and **Zoltan Kunszt**

- Constructing a NLO generator for colliders
  - Ordered vs Dressed amplitudes.
- Dressed Formalism at one-loop
  - Algorithmic formulation
- Numerical implementation of virtual corrections
  - Speed, Stability and Convergence

Radcor, Ascona, 28 October,2009

Not ROCKET

Constructing a NLO Generator

- There are many LO generators (e.g. Alpgen, Madgraf, Amegic, Comix, …).

- These generators are important tools for phenomenology and experimenters.

- They give a qualitative understanding of prediction uncertainties.

- Adding NLO corrections will give us a more quantitative understanding of the uncertainties in the predictions due to the α_S expansion

Traditional method for numerical evaluation of virtual correction (within our framework):

- Separate color factor from amplitude to obtain ordered amplitudes

Berends, Giele (1987)
Del Duca, Dixon, Maltoni (2000)

$$\mathcal{M}^{(0,1)}(1,2\ldots,n) \sim \sum_{P(23\cdots n)} Tr\left(F^{a_1}F^{a_2}\cdots F^{a_n}\right) m^{(0,1)}(1,2,\ldots,n)$$

Maltoni, Paul, Stelzer, Willenbrock (2003)

$$\begin{aligned}
\mathcal{M}^{(0)}(\mathbf{g_1},\ldots,\mathbf{g_n}) &= \sum_{P(2\cdots n)} A^{(0)}{}^{i_1\cdots i_n}_{j_1\cdots j_n}(g_1^{\lambda_1},\ldots,g_n^{\lambda_n}) \\
&= T^{a_1}_{i_1 j_1}\cdot T^{a_n}_{i_n j_n} \sum_{P(2\cdots n)} \mathrm{Tr}\Big(F^{a_1}\cdots F^{a_n}\Big) \times m^{(0)}(g_1^{\lambda_1},\ldots,g_n^{\lambda_n}) \\
&\sim \frac{1}{2}\sum_{P(2\cdots n)} \Big(\delta^{i_1}_{j_2}\delta^{i_2}_{j_3}\cdots\delta^{i_{n-1}}_{j_n}\delta^{i_n}_{j_1} + (-1)^n\delta^{i_n}_{j_{n-1}}\delta^{i_{n-1}}_{j_{n-2}}\cdots\delta^{i_2}_{j_1}\delta^{i_1}_{j_n}\Big) \times m^{(0)}(g_1^{\lambda_1},\ldots,g_n^{\lambda_n}) \\
&= \sum_{P(2\cdots n)} \delta^{i_1}_{j_2}\delta^{i_2}_{j_3}\cdots\delta^{i_{n-1}}_{j_n}\delta^{i_n}_{j_1} \times m^{(0)}(g_1^{\lambda_1},\ldots,g_n^{\lambda_n})\ , \qquad (2
\end{aligned}$$

$$\mathcal{M}^{(1)}(\mathbf{g_1},\ldots,\mathbf{g_n}) = \sum_{P(2\cdots n)} A^{(1)}{}^{i_1\cdots i_n}_{j_1\cdots j_n}(g_1^{\lambda_1},\ldots,g_n^{\lambda_n})$$

$$= \sum_{P(1\cdots n-1)} \left[N_c\Delta_{1\cdots n} + \sum_{k=1}^{\mathrm{int}(n/2)}\sum_{m_1=1}^{n-k+1}\cdots\sum_{m_k=m_{k-1}+1}^{n}(-1)^k\Delta_{m_1\cdots m_k}\Delta_{1\cdots\cancel{m}_1\cdots\cancel{m}_k\cdots n}\right] m^{(1)}(12\cdots n)$$

Bern, Dixon, Dunbar, Kosower (1995)
Britto, Cachazo, Feng (2002)

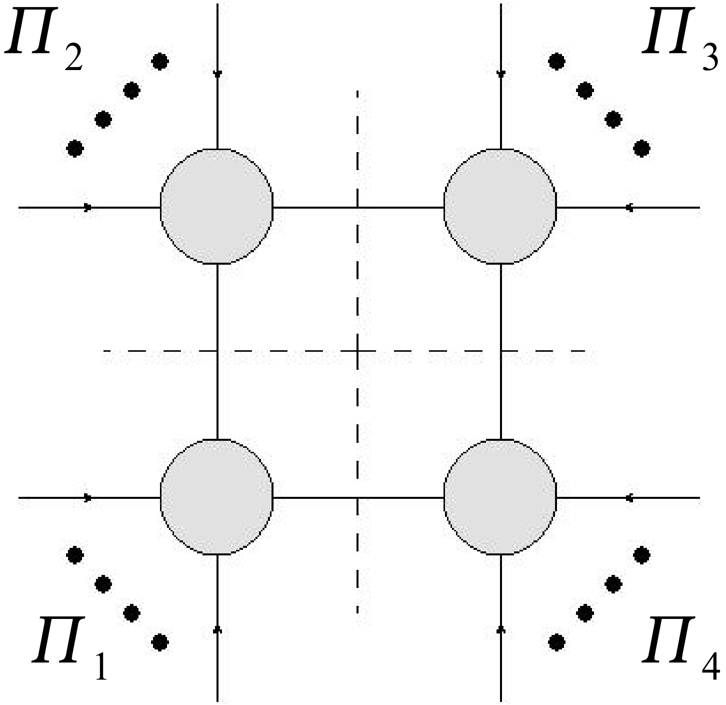
Ossola, Papadopoulos, Pittau (2007)

- Use generalized unitarity combined with OPP parametric integration to build a purely numerical implementation of generalized unitarity

Ellis, Giele, Kunszt (2007)

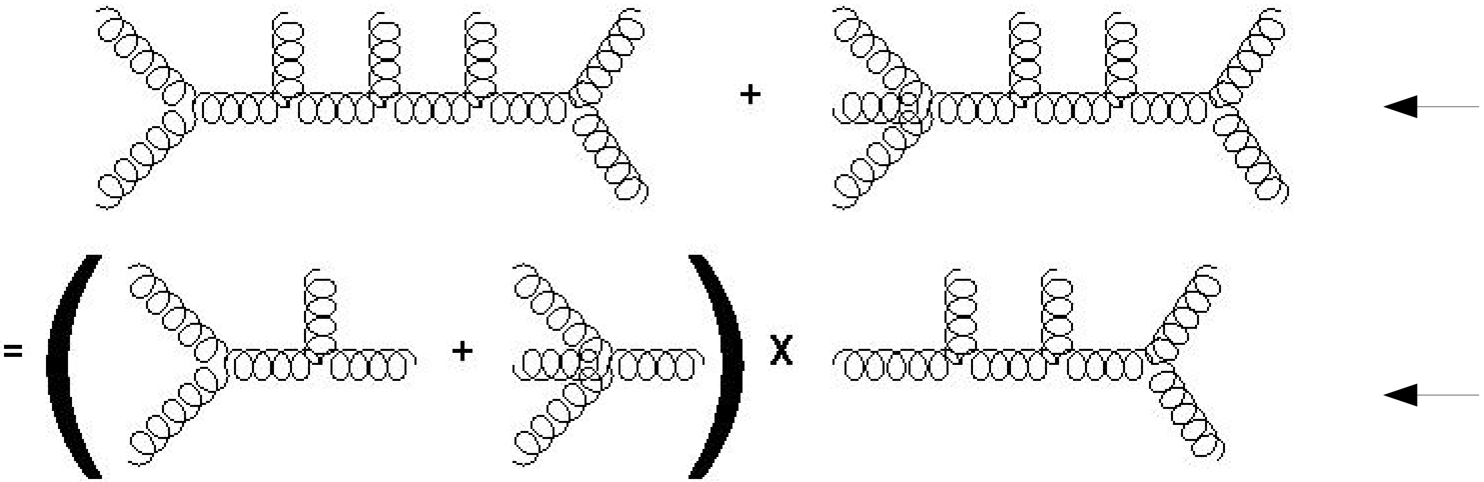
- We use ***D=5*** unitarity cuts to calculate the ***D***-dimensional coefficients of the master integrals

Giele, Kunszt, Melnikov (2008)



- Most salient feature is factorization of the Feynman graphs in on-shell tree-level blobs!
- This beats the factorial scaling of the amplitude
- There are n^D cuts per ordered amplitude
- The ordered tree-level blobs are calculated using Berends-Giele recursion relations, scaling as $n^{V_{max}}$
- The resulting scaling of the calculation of the ordered one-loop amplitude is $n^{D+V_{max}} = n^9$
- Crucial in the polynomial scaling of the algorithm is the recursion relation, why is this not a factorial algorithm?

- Again, this is based on factorization of Feynman diagrams into common factors:

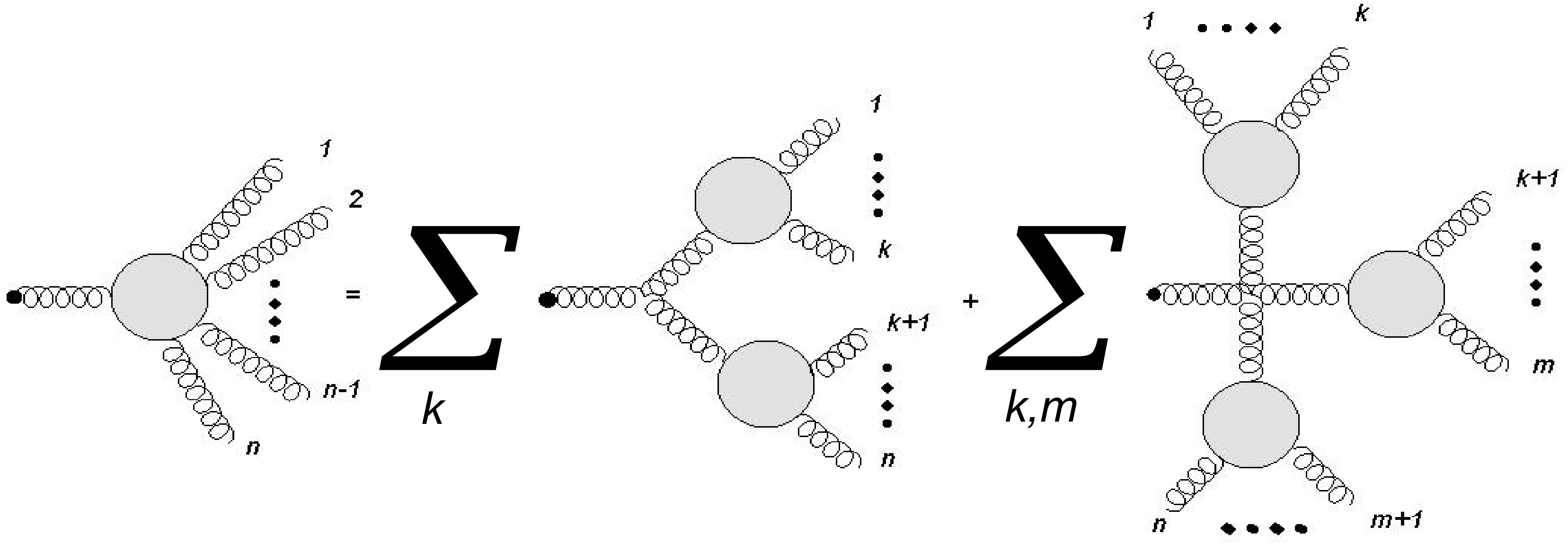


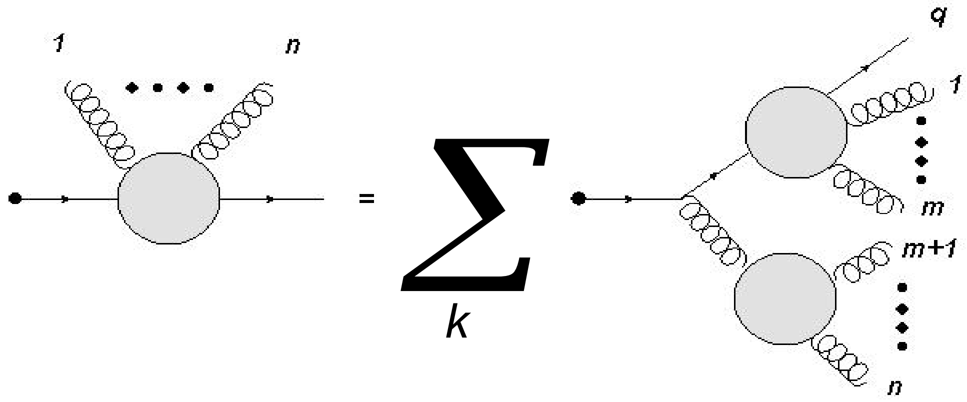
9 vertex operations

6 vertex operations

Berends, Giele (1988)

- Formalize into recursion relations which maximizes the factorization.
- Reduces algorithmic scaling from double factorial to polynomial for ordered amplitudes.





To calculate the virtual corrections to ***n***-gluon scattering in the leading color approximation we are done now:

$$\mathcal{M}^{(0,1)}(1,2\ldots,n) \sim \sum_{P(23\cdots n)} Tr\left(F^{a_1}F^{a_2}\cdots F^{a_n}\right) m^{(0,1)}(1,2,\ldots,n)$$

$$\begin{aligned}\Big\langle m(12\cdots n)\Big\rangle^2 &= Tr\left(T^{a_1}T^{a_2}\cdots T^{a_n}\right) m(12\cdots n) \times \left(\sum_{P(2\cdots n)} Tr\left(T^{a_n}\cdots T^{a_2}T^{a_1}\right) m(n\cdots 21)^\dagger\right) \\ &= (N_c^2-1)N_c^{n-2}\left(\Big|m(12\cdots n)\Big|^2 + \mathcal{O}\left(\frac{1}{N_c^2}\right)\right) .\end{aligned} \tag{C.2}$$

$$\begin{aligned}d\sigma(gg\to(n-2)g) &= \frac{1}{(n-2)!} d\,PS(p_1p_2\to p_2\cdots p_n)\Big|\mathcal{M}(g_1(p_1),g_2(p_2),\ldots,g_n(p_n))\Big|^2 \\ &= (n-1)d\,PS(p_1p_2\to p_2\cdots p_n)\Big\langle m(g_1(p_{\sigma_1})g_2(p_{\sigma_2})\cdots g_n(p_{\sigma_n}))\Big\rangle^2\end{aligned}$$

- The factorial permutation sum over orderings is irrelevant in the leading color approximation . → Polynomial scaling
- Without any color approximations the permutation sum over orderings needs to be performed → Factorial scaling is back

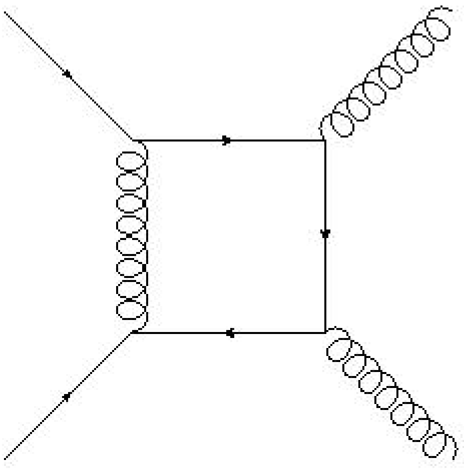
Including quarks, the non-leading color contributions add more complexity

Del Duca, Dixon, Maltoni (2000)

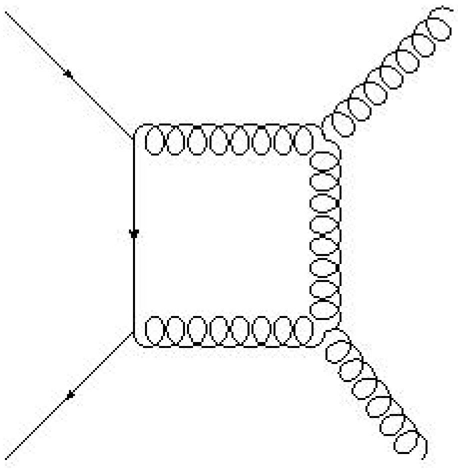
$$\mathcal{M}^{(1)}(q;1,\ldots,n;\bar{q}) \sim \sum_{k=2}^{n} \sum_{P(1\cdots n)} (T^y T^{a_1}\cdots T^{a_k} T^x)_{ij}\, (F^{a_{k+1}}\cdots F^{a_n})_{xy}\, m^{(1)}(q,1,\ldots,k,q,k+1,\ldots,n)$$

Bern, Dixon, Dunbar, Kosower (1994)

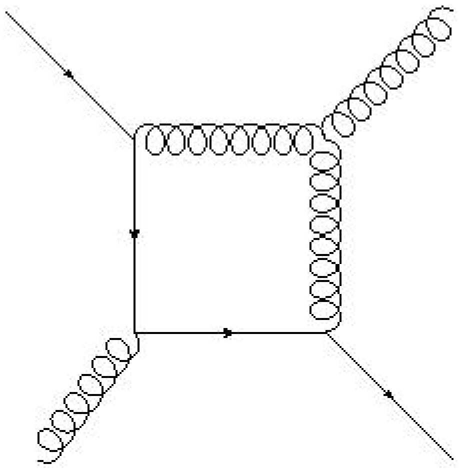
- Ordered amplitudes break up in primitive amplitudes each with its own color factor
- This is understood through the internal flavor structure of the loop
- From an algorithmic point of view this adds complications (have to keep track of these many different types primitive amplitudes)
- (At leading color again no issue as only primitive amplitude contributes.)



$(T^y T^{a_1} T^{a_2} T^x)_{ij}\,\delta_{xy}$



$(T^y T^x)_{ij}(F^{a_1} F^{a_2})_{xy}$



$(T^y T^{a_1} T^x)_{ij}\,(F^{a_2})_{xy}$

Ellis, Giele, Kunszt, Melnikov, Zanderighi (2008)

Adding more quarks results in more “chaos”: Eg W+qqb+QQb+g

$$\mathcal{B}^{\rm tree}(1_{\bar q},2_q,3_{\bar Q},4_Q,5_g) = g^3\Bigg[(T^{a_5})_{i_4}^{\bar\imath_1}\delta_{i_2}^{\bar\imath_3}B_{7;1}^{\rm tree} + \frac{1}{N_c}(T^{a_5})_{i_2}^{\bar\imath_1}\delta_{i_4}^{\bar\imath_3}B_{7;2}^{\rm tree}$$

$$+ (T^{a_5})_{i_2}^{\bar\imath_3}\delta_{i_4}^{\bar\imath_1}B_{7;3}^{\rm tree} + \frac{1}{N_c}(T^{a_5})_{i_4}^{\bar\imath_3}\delta_{i_2}^{\bar\imath_1}B_{7;4}^{\rm tree}\Bigg],$$

$$\mathcal{B}^{\text{1-loop}}(1_{\bar q},2_q,3_{\bar Q},4_Q,5_g) = g^5\Bigg[N_c(T^{a_5})_{i_4}^{\bar\imath_1}\delta_{i_2}^{\bar\imath_3}B_{7;1} + (T^{a_5})_{i_2}^{\bar\imath_1}\delta_{i_4}^{\bar\imath_3}B_{7;2}$$

$$+ N_c(T^{a_5})_{i_2}^{\bar\imath_3}\delta_{i_4}^{\bar\imath_1}B_{7;3} + (T^{a_5})_{i_4}^{\bar\imath_3}\delta_{i_2}^{\bar\imath_1}B_{7;4}\Bigg].$$

$$B_{7;1}^{[1],a} = \left(1-\frac{1}{N_c^2}\right)A_L^{[1],a}(1_{\bar q},2_q,3_{\bar Q},4_Q,5_g) - \frac{1}{N_c^2}\Big(-A_L^{[1],a}(1_{\bar q},5_g,2_q,3_{\bar Q},4_Q) \qquad (5.6)$$

$$-A_L^{[1],a}(1_{\bar q},5_g,2_q,4_Q,3_{\bar Q}) - A_L^{[1],a}(1_{\bar q},2_q,5_g,3_{\bar Q},4_Q) - A_L^{[1],a}(1_{\bar q},2_q,5_g,4_Q,3_{\bar Q})$$

$$-A_L^{[1],a}(1_{\bar q},2_q,3_{\bar Q},5_g,4_Q) - A_L^{[1],a}(1_{\bar q},2_q,4_Q,5_g,3_{\bar Q}) + A_L^{[1],a}(1_{\bar q},2_q,4_Q,3_{\bar Q},5_g)\Big),$$

$$B_{7;2}^{[1],a} = +A_L^{[1],a}(1_{\bar q},2_q,5_g,4_Q,3_{\bar Q}) - A_L^{[1],a}(1_{\bar q},2_q,4_Q,5_g,3_{\bar Q}) + A_L^{[1],a}(1_{\bar q},2_q,4_Q,3_{\bar Q},5_g)$$

$$-\frac{1}{N_c^2}\Big(A_L^{[1],a}(1_{\bar q},5_g,2_q,3_{\bar Q},4_Q) + A_L^{[1],a}(1_{\bar q},5_g,2_q,4_Q,3_{\bar Q})\Big), \qquad (5.7)$$

$$B_{7;3}^{[1],a} = \left(1-\frac{1}{N_c^2}\right)A_L^{[1],a}(1_{\bar q},2_q,5_g,3_{\bar Q},4_Q) - \frac{1}{N_c^2}\Big(-A_L^{[1],a}(1_{\bar q},5_g,2_q,3_{\bar Q},4_Q) \qquad (5.8)$$

$$-A_L^{[1],a}(1_{\bar q},5_g,2_q,4_Q,3_{\bar Q}) - A_L^{[1],a}(1_{\bar q},2_q,3_{\bar Q},5_g,4_Q) - A_L^{[1],a}(1_{\bar q},2_q,4_Q,5_g,3_{\bar Q})$$

$$-A_L^{[1],a}(1_{\bar q},2_q,3_{\bar Q},4_Q,5_g) - A_L^{[1],a}(1_{\bar q},2_q,4_Q,3_{\bar Q},5_g) + A_L^{[1],a}(1_{\bar q},2_q,5_g,4_Q,3_{\bar Q})\Big),$$

$$B_{7;4}^{[1],a} = -A_L^{[1],a}(1_{\bar q},5_g,2_q,4_Q,3_{\bar Q}) - A_L^{[1],a}(1_{\bar q},2_q,5_g,4_Q,3_{\bar Q}) - A_L^{[1],a}(1_{\bar q},2_q,4_Q,3_{\bar Q},5_g)$$

$$-\frac{1}{N_c^2}\Big(A_L^{[1],a}(1_{\bar q},2_q,4_Q,5_g,3_{\bar Q}) + A_L^{[1],a}(1_{\bar q},2_q,3_{\bar Q},5_g,4_Q)\Big). \qquad (5.9)$$

$$B_{7;1}^{[1],b} = \frac{1}{N_c^2}A_L^{[1],b}(1_{\bar q},5_g,4_Q,3_{\bar Q},2_q),$$

$$B_{7;2}^{[1],b} = -\frac{1}{N_c^2}\Big(A_L^{[1],b}(1_{\bar q},5_g,4_Q,3_{\bar Q},2_q) + A_L^{[1],b}(1_{\bar q},4_Q,5_g,3_{\bar Q},2_q) + A_L^{[1],b}(1_{\bar q},4$$

$$B_{7;3}^{[1],b} = \frac{1}{N_c^2}A_L^{[1],b}(1_{\bar q},4_Q,3_{\bar Q},5_g,2_q),$$

$$B_{7;4}^{[1],b} = -A_L^{[1],b}(1_{\bar q},5_g,4_Q,3_{\bar Q},2_q) - A_L^{[1],b}(1_{\bar q},4_Q,3_{\bar Q},5_g,2_q) - A_L^{[1],b}(1_{\bar q},4_Q,3_{\bar Q}$$

$$-\left(1-\frac{1}{N_c^2}\right)A_L^{[1],b}(1_{\bar q},4_Q,5_g,3_{\bar Q},2_q).$$

$$B_{7;1}^{[1],c} = \frac{1}{N_c^2}A_L^{[1],c}(1_{\bar q},5_g,4_Q,3_{\bar Q},2_q),$$

$$B_{7;2}^{[1],c} = -A_L^{[1],c}(1_{\bar q},5_g,4_Q,3_{\bar Q},2_q) - A_L^{[1],c}(1_{\bar q},4_Q,5_g,3_{\bar Q},2_q) - A_L^{[1$$

$$-\left(1-\frac{1}{N_c^2}\right)A_L^{[1],c}(1_{\bar q},4_Q,3_{\bar Q},2_q,5_g),$$

$$B_{7;3}^{[1],c} = \frac{1}{N_c^2}A_L^{[1],c}(1_{\bar q},4_Q,3_{\bar Q},5_g,2_q),$$

$$B_{7;4}^{[1],c} = -\frac{1}{N_c^2}\Big(A_L^{[1],c}(1_{\bar q},5_g,4_Q,3_{\bar Q},2_q) + A_L^{[1],c}(1_{\bar q},4_Q,3_{\bar Q},5_g,2_q) + A_L^{[1$$

- For calculating virtual corrections to specific processes the use of primitive amplitudes is convenient.
- The primitive amplitude decomposition is not well suited for algorithmic evaluation generic multi-parton processes.
- (In the large color limit the primitive amplitude description is greatly simplified and suitable of algorithmic implementation.)

- Following the development path of the LO generators we introduce
  (1) MC sampling over external quantum numbers (color, helicity,...)
  (2) Dressed recursive algorithms for tree-level blobs

$$d\,\sigma_{LO}(f_1 f_2 \to f_3 \cdots f_n) = \frac{W_S}{N_{\rm event}} \times \sum_{r=1}^{N_{\rm event}} d\,PS^{(r)}(K_1 K_2 \to K_3 \cdots K_n) \left| \mathcal{M}^{(0)} \left(\mathbf{f}_1^{(r)}, \mathbf{f}_2^{(r)}, \ldots, \mathbf{f}_n^{(r)} \right) \right|^2$$

$$d\,\sigma^{(V)}(f_1 f_2 \to f_3 \cdots f_n) = \frac{W_S}{N_{\rm event}} \times \sum_{k=1}^{N_{\rm event}} d\,PS^{(k)}(K_1 K_2 \to K_3 \cdots K_n)$$
$$2\Re \left(\mathcal{M}^{(0)} \left(\mathbf{f}_1^{(k)}, \ldots, \mathbf{f}_n^{(k)} \right)^{\dagger} \times \mathcal{M}^{(1)} \left(\mathbf{f}_1^{(k)}, \ldots, \mathbf{f}_n^{(k)} \right) \right) ,$$

Van Hameren, Papadopoulos, Pittau (2009)

- The HELAC group uses MC sampling over color/helicity within the primitive amplitude decomposition framework (following their LO sampling using ordered amplitudes)
- We will call this technique in what follows “ordered sampling”
- Ordered sampling is efficient because lots of color factors are zero
- We are interested in virtual corrections, so the simpler LO color factors are the relevant quantities to examine

$$
\begin{aligned}
\mathcal{M}^{(0)}(\mathbf{g_1},\ldots,\mathbf{g_n}) &= \sum_{P(2\cdots n)} A^{(0)}{}^{i_1\cdots i_n}_{j_1\cdots j_n}(g_1^{\lambda_1},\ldots,g_n^{\lambda_n}) \\
&= T^{a_1}_{i_1j_1}\cdot T^{a_n}_{i_nj_n} \sum_{P(2\cdots n)} \mathrm{Tr}\Big(F^{a_1}\cdots F^{a_n}\Big)\times m^{(0)}(g_1^{\lambda_1},\ldots,g_n^{\lambda_n}) \\
&\sim \frac{1}{2}\sum_{P(2\cdots n)} \Big(\delta^{i_1}_{j_2}\delta^{i_2}_{j_3}\cdots\delta^{i_{n-1}}_{j_n}\delta^{i_n}_{j_1} + (-1)^n\delta^{i_n}_{j_{n-1}}\delta^{i_{n-1}}_{j_{n-2}}\cdots\delta^{i_2}_{j_1}\delta^{i_1}_{j_n}\Big)\times m^{(0)}(g_1^{\lambda_1},\ldots,g_n^{\lambda_n}) \\
&= \sum_{P(2\cdots n)} \delta^{i_1}_{j_2}\delta^{i_2}_{j_3}\cdots\delta^{i_{n-1}}_{j_n}\delta^{i_n}_{j_1}\times m^{(0)}(g_1^{\lambda_1},\ldots,g_n^{\lambda_n})\ , \qquad (2
\end{aligned}
$$

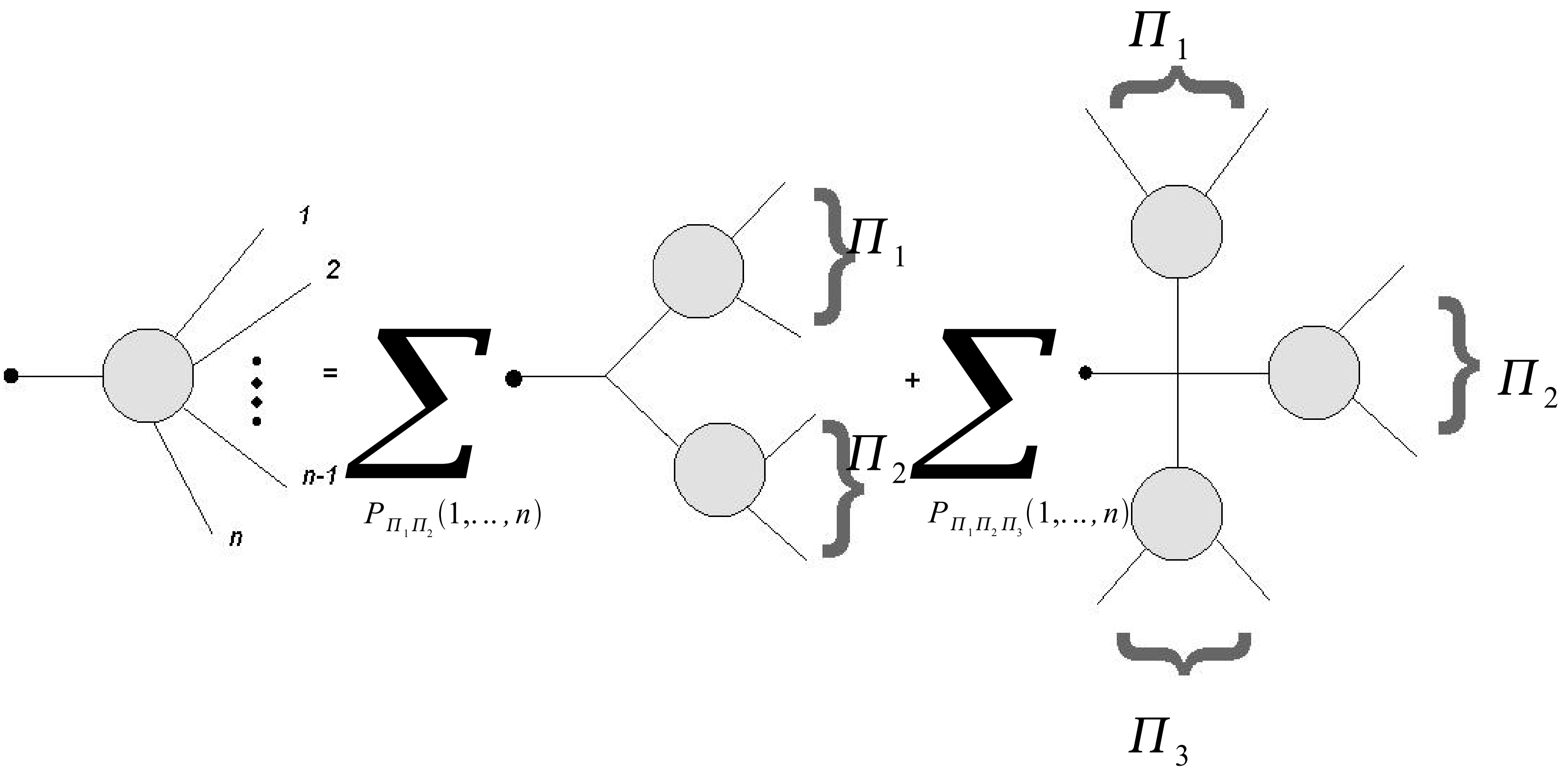
scattering	Naive	Conserved	Non-Zero
$2 \to 2$	6,561	639	378
$2 \to 3$	59,049	4,653	3,180
$2 \to 4$	531,441	35,169	27,240
$2 \to 5$	4,782,969	272,835	231,672
$2 \to 6$	43,046,721	2,157,759	1,949,178

- Because colors are explicit we can calculate the color weight for each ordering
- ~95% of the weight are zero
- We only have to calculate the non-zero color weight ordering at tree-level and one-loop to get the virtual corrections
- However, we still have factorial scaling of the algorithm. Not desirable...

Duhr, Hoche, Maltoni (2006)

- To beat the factorial we follow the method of COMIX by employing dressed recursion relations

Gleisberg, Hoche (2008)


1
2
n-1
n
=
$\sum$
$P_{\Pi_1 \Pi_2}(1,\ldots,n)$
Π_1
Π_2
+
$\sum$
$P_{\Pi_1 \Pi_2 \Pi_3}(1,\ldots,n)$
Π_1
Π_2
Π_3

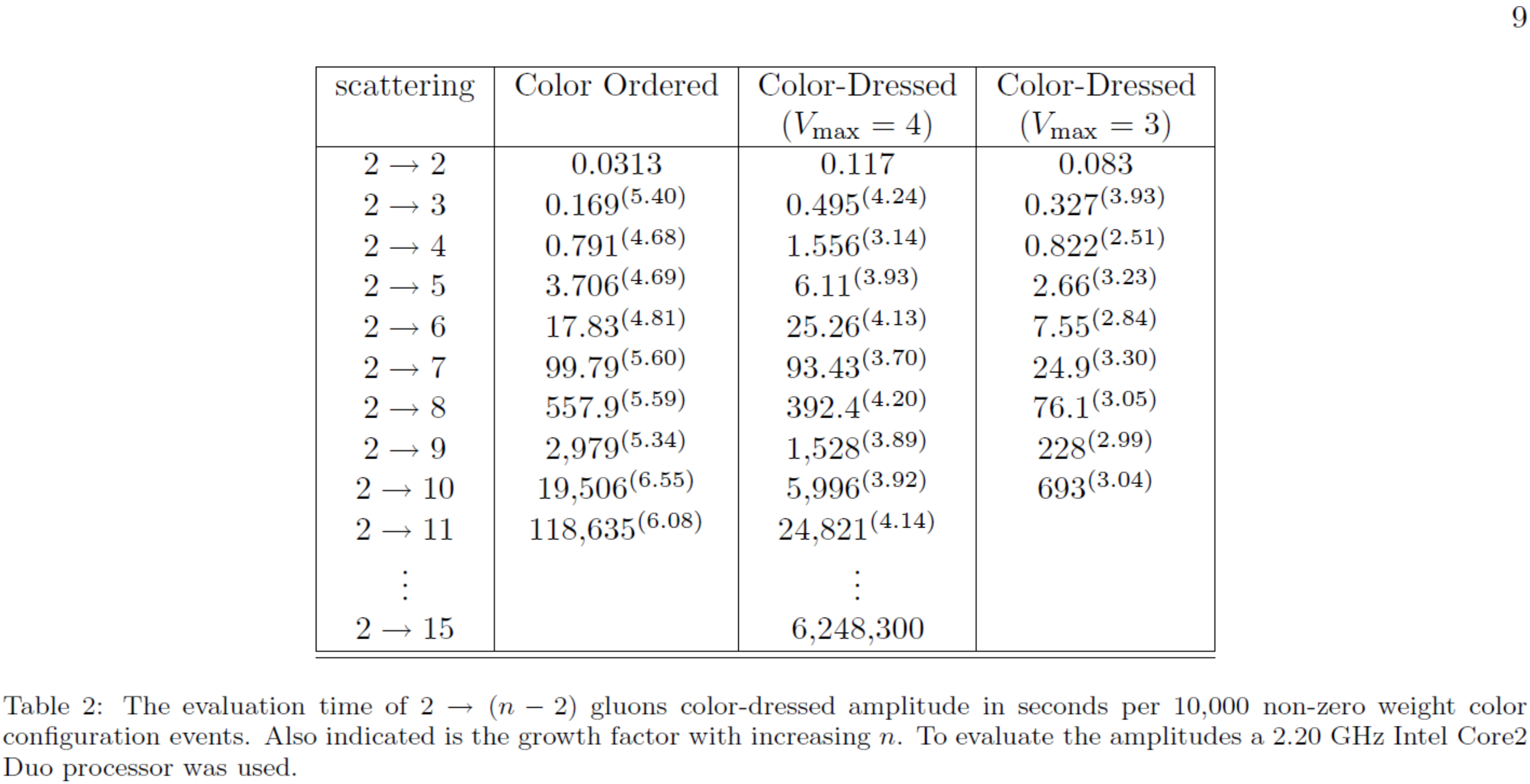
- Fully specified external state; Particle content irrelevant: gluon current, quark current, W current,...
- Calculates full amplitude
- Add in all vertices and set a vertex like (q,g,g) to zero.
- Algorithmic implementation is “blind” to the particle content.
- Scales as $(V_{max})^n$ instead of the ordered amplitude scaling $n^{V_{max}}$
- Algorithm scales exponential (it beats the factorial).

$$\mathcal{M}^{(0)}\left(\mathbf{f}_1,\ldots,\mathbf{f}_n\right)=\sum_{\mathbf{g}} J_{\mathbf{g}}\left(\mathbf{f}_1,\ldots,\mathbf{f}_{n-1}\right)J^{\mathbf{g}}\left(\mathbf{f}_n\right)\Big|_{K_1+\cdots+K_n=0}$$

$$J_{\mathbf{g}}(\mathbf{f}_1,\ldots,\mathbf{f}_n)=\sum_{k=2}^{V_{\mathrm{max}}-1}\ \sum_{P_{\pi_1\cdots\pi_k}(1\cdots n)}^{\mathcal{S}_2(n,k)} P_{\mathbf{g}}\left[D\left[J(\mathbf{f}_{\pi_1}),\ldots,J(\mathbf{f}_{\pi_k})\right]\right]$$

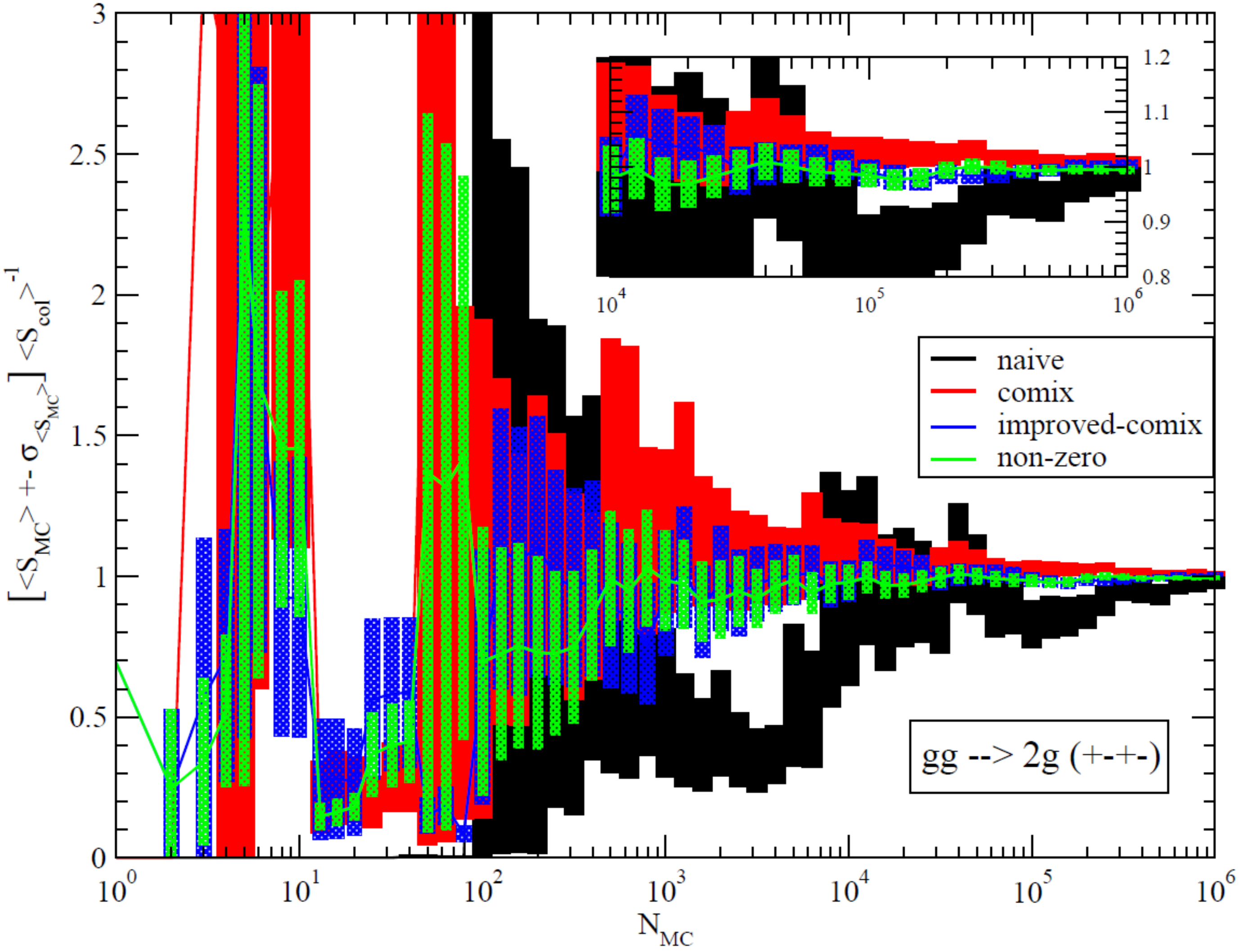
- These formula specify the algorithm, which can be implemented easily in any object oriented language
- Gives an automated tree-level generator (COMIX) for any Feynman graph based theory.
- Simply add all vertices and choose the external sources to produce the matrix element squared.

- We implemented the dressed recursion relation (with the appropriate extensions needed for use in the ***D*-**dimensional generalized unitarity framework)
- Some results:



scattering	Color Ordered	Color-Dressed ($V_{\max} = 4$)	Color-Dressed ($V_{\max} = 3$)
$2 \to 2$	0.0313	0.117	0.083
$2 \to 3$	$0.169^{(5.40)}$	$0.495^{(4.24)}$	$0.327^{(3.93)}$
$2 \to 4$	$0.791^{(4.68)}$	$1.556^{(3.14)}$	$0.822^{(2.51)}$
$2 \to 5$	$3.706^{(4.69)}$	$6.11^{(3.93)}$	$2.66^{(3.23)}$
$2 \to 6$	$17.83^{(4.81)}$	$25.26^{(4.13)}$	$7.55^{(2.84)}$
$2 \to 7$	$99.79^{(5.60)}$	$93.43^{(3.70)}$	$24.9^{(3.30)}$
$2 \to 8$	$557.9^{(5.59)}$	$392.4^{(4.20)}$	$76.1^{(3.05)}$
$2 \to 9$	$2{,}979^{(5.34)}$	$1{,}528^{(3.89)}$	$228^{(2.99)}$
$2 \to 10$	$19{,}506^{(6.55)}$	$5{,}996^{(3.92)}$	$693^{(3.04)}$
$2 \to 11$	$118{,}635^{(6.08)}$	$24{,}821^{(4.14)}$	
$\vdots$		$\vdots$	
$2 \to 15$		6,248,300	

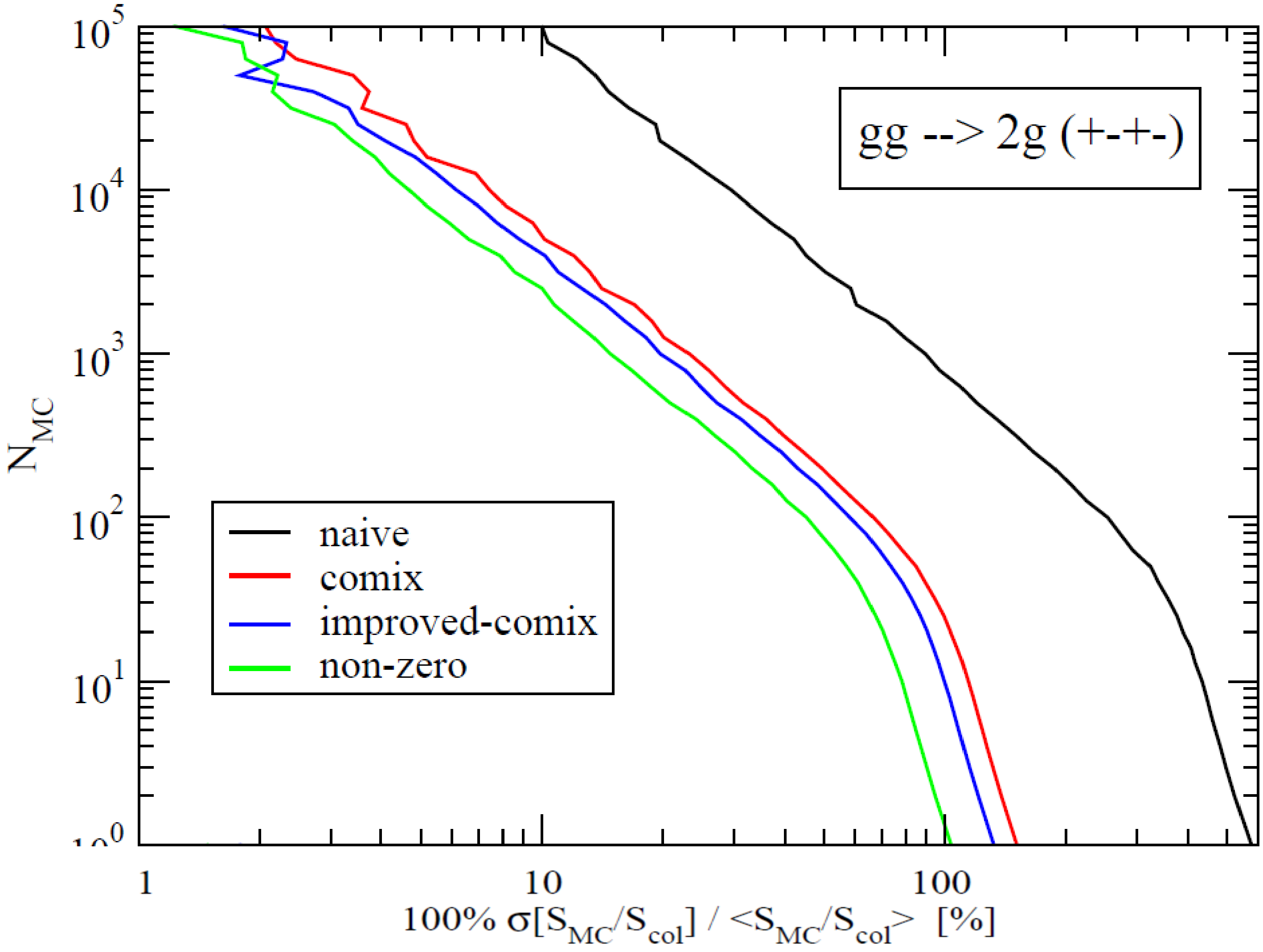
Table 2: The evaluation time of $2 \to (n-2)$ gluons color-dressed amplitude in seconds per 10,000 non-zero weight color configuration events. Also indicated is the growth factor with increasing n. To evaluate the amplitudes a 2.20 GHz Intel Core2 Duo processor was used.



$$\langle S_{\mathrm{MC}}\rangle = \frac{1}{N_{\mathrm{MC}}}\sum_{k=1}^{N_{\mathrm{MC}}}\left|\mathcal{M}^{(0)}(\mathbf{g}_1^{(\mathbf{k})},\ldots,\mathbf{g}_{\mathbf{N}}^{(\mathbf{k})})\right|^2$$

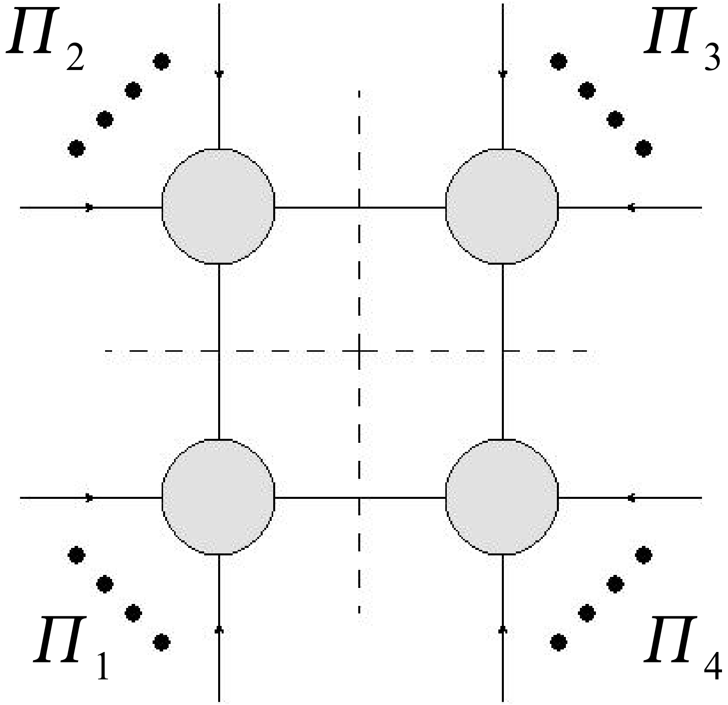
$$\sigma_{\langle S_{\mathrm{MC}}\rangle} = \sqrt{\frac{1}{N_{\mathrm{MC}}}\sum_{k=1}^{N_{\mathrm{MC}}}\left(\left|\mathcal{M}^{(0)}(\mathbf{g}_1^{(\mathbf{k})},\ldots,\mathbf{g}_{\mathbf{N}}^{(\mathbf{k})})\right|^2-\langle S_{\mathrm{MC}}\rangle\right)^2}$$

$$\langle S_{\mathrm{col}}\rangle = \frac{1}{N_{\mathrm{MC}}}\sum_{k=1}^{N_{\mathrm{MC}}}\sum_{i_1\cdots i_N=1}^{3}\sum_{j_1\cdots j_N=1}^{3}\left|\mathcal{M}^{(0)}(\mathbf{g}_1^{(\mathbf{k})},\ldots,\mathbf{g}_{\mathbf{N}}^{(\mathbf{k})})\right|^2$$



We can summarize the previous graph in a more informative graph or even better in quantitative numbers:

$$\begin{aligned}
\text{Naive} &: \quad \frac{\sigma\left(S_{\mathrm{MC}}/S_{\mathrm{col}}\right)}{\langle S_{\mathrm{MC}}/S_{\mathrm{col}}\rangle} = 23.22 \times N_{\mathrm{MC}}^{-0.47} \\
\text{Conserved} &: \quad \frac{\sigma\left(S_{\mathrm{MC}}/S_{\mathrm{col}}\right)}{\langle S_{\mathrm{MC}}/S_{\mathrm{col}}\rangle} = 7.44 \times N_{\mathrm{MC}}^{-0.51} \\
\text{Non-Zero} &: \quad \frac{\sigma\left(S_{\mathrm{MC}}/S_{\mathrm{col}}\right)}{\langle S_{\mathrm{MC}}/S_{\mathrm{col}}\rangle} = 3.53 \times N_{\mathrm{MC}}^{-0.51} \; .
\end{aligned}$$



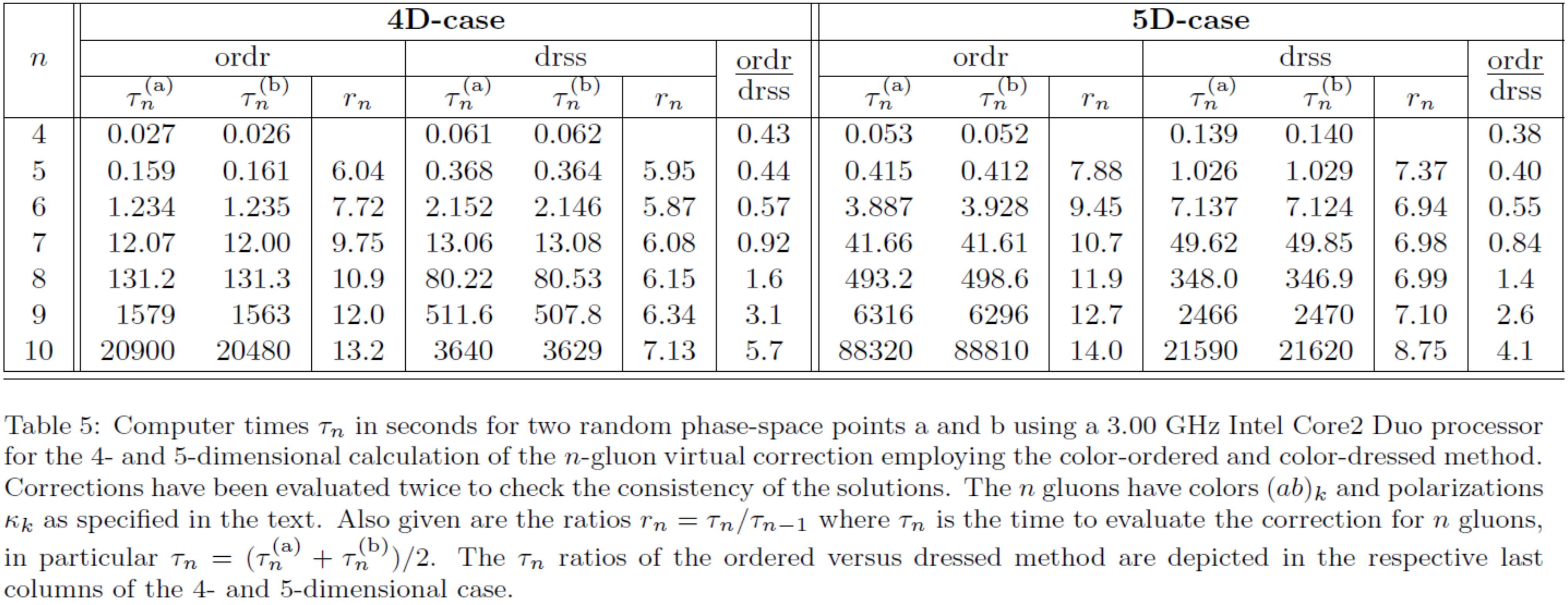
- *D*-dimensional Generalized Unitarity + OPP gives a fully numerical implementation of one-loop calculations
- The physics input is tree-level blobs.
- The one-loop amplitude inherits the color dressing from tree-level recursion.
- Requires color summation over internal lines (color ordered sampling does not).
- How fast is it compared the color ordered evaluation with its more analytic treatment of color?
- Below is the full algorithmic definition which again has exponential scaling
- The numerical implementation is more challenging, but my younger collaborator programmed it without much problems

$$\mathcal{A}^{(1)}\left(\mathbf{f_1},\ldots,\mathbf{f_n}|\ell\right) = \sum_{k=1}^{C_{\max}} \sum_{RP_{\pi_1\cdots\pi_k}(12\cdots n)}^{\max\left(1,\frac{1}{2}(k-1)!\times\mathcal{S}_2(n,k)\right)} \sum_{g_{\pi_1},\ldots,g_{\pi_k}} \frac{\mathcal{P}_k\left(\{c_{g_{\pi_1}\cdots g_{\pi_k}}\}|\ell\right)}{d_{\Pi_1}^{(g_1)}(\ell)d_{\Pi_2}^{(g_2)}(\ell)\cdots d_{\Pi_k}^{(g_k)}(\ell)}$$

$$\begin{aligned}
\mathcal{P}_c\left(\{c_{g_{\pi_1}\cdots g_{\pi_c}}\}\mid \ell_{\Pi_1\cdots\Pi_c}\right) &= \mathrm{Res}_{g_{\pi_1}\cdots g_{\pi_c}}\left(\mathcal{A}^{(1)}\left(\mathbf{f_1},\ldots,\mathbf{f_n}\mid \ell_{\Pi_1\cdots\Pi_c}\right)\right) \\
&- \sum_{m=c+1}^{C_{\max}} \sum_{RP_{\pi_1,\ldots,\pi_m}(1\ldots n)} \sum_{g_{\pi_{c+1}}\cdots g_{\pi_m}} \frac{\mathcal{P}_m(\{c_{g_{\pi_1}\cdots g_{\pi_m}}\}\mid \ell_{\Pi_1\cdots\Pi_c})}{d_{\Pi_{(c+1)}}^{(g_{c+1})}(\ell_{\Pi_1\cdots\Pi_c})\cdots d_{\Pi_m}^{(g_m)}(\ell_{\Pi_1\cdots\Pi_c})}
\end{aligned}$$

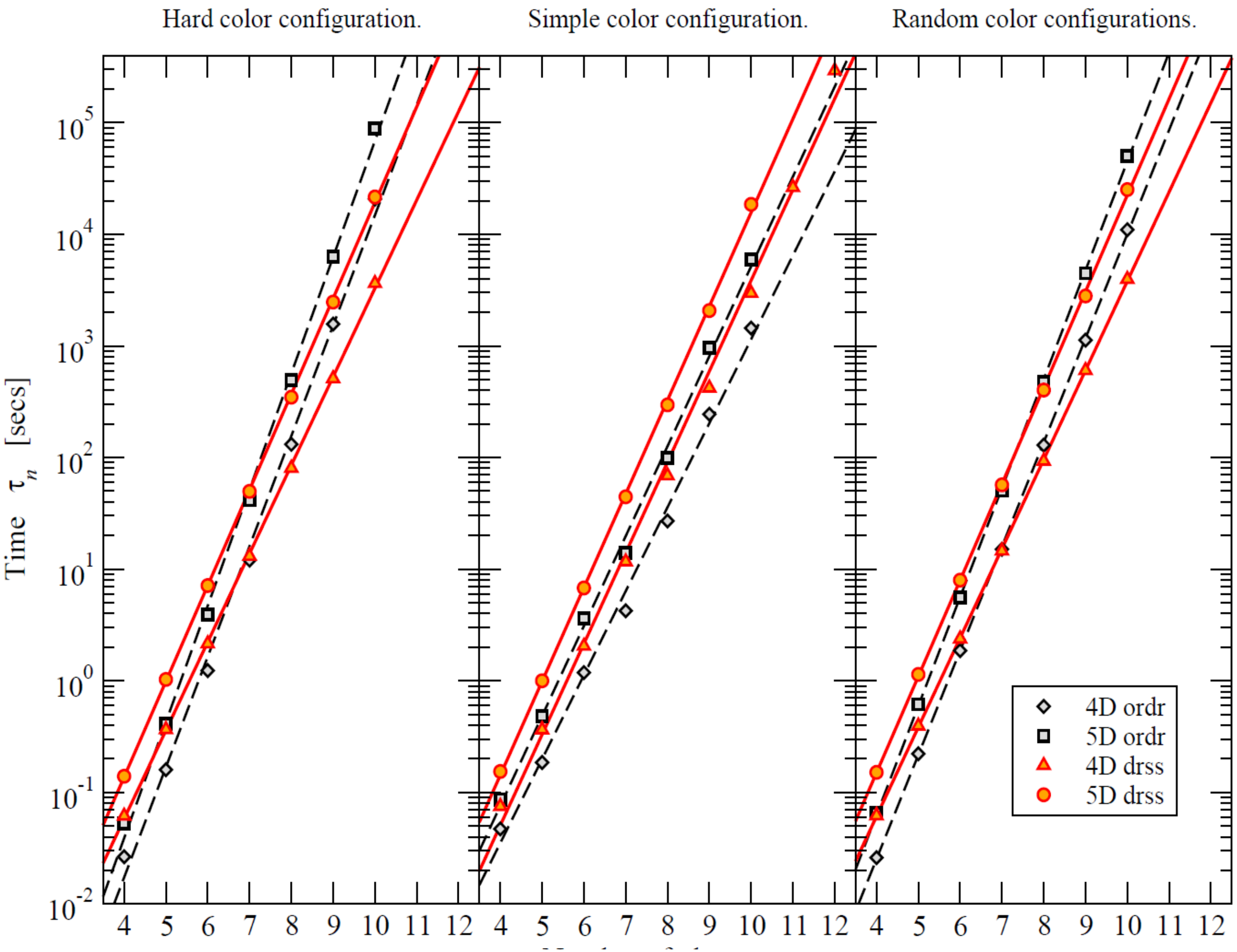
$$\begin{aligned}
\mathrm{Res}_{g_{\pi_1}\cdots g_{\pi_c}}\left(\mathcal{A}^{(1)}\left(\mathbf{f_1},\ldots,\mathbf{f_n}\mid \ell_{\Pi_1\cdots\Pi_c}\right)\right) &= \left[d_{\Pi_1}^{(g_1)}\times\cdots\times d_{\Pi_c}^{(g_c)}\times\mathcal{A}^{(1)}\left(\mathbf{f_1},\ldots,\mathbf{f_n}|\ell\right)\right]_{\ell=\ell_{\Pi_1\cdots\Pi_c}} \\
&= \sum_{\mathbf{g_1}\cdots\mathbf{g_{k+1}}} \prod_{k=1}^{c}\left[\mathcal{M}^{(0)}\left((\mathbf{g_k})^{\dagger},\{\mathbf{f}\}_{\pi_k},\mathbf{g_{k+1}}\right)\right] ,
\end{aligned}$$

Evaluation time for single phase space point evaluation of the *n*-gluon virtual correction:



n	4D-case							5D-case						
	ordr			drss			$\frac{\text{ordr}}{\text{drss}}$	ordr			drss			$\frac{\text{ordr}}{\text{drss}}$
	$\tau_n^{(a)}$	$\tau_n^{(b)}$	r_n	$\tau_n^{(a)}$	$\tau_n^{(b)}$	r_n		$\tau_n^{(a)}$	$\tau_n^{(b)}$	r_n	$\tau_n^{(a)}$	$\tau_n^{(b)}$	r_n	
4	0.027	0.026		0.061	0.062		0.43	0.053	0.052		0.139	0.140		0.38
5	0.159	0.161	6.04	0.368	0.364	5.95	0.44	0.415	0.412	7.88	1.026	1.029	7.37	0.40
6	1.234	1.235	7.72	2.152	2.146	5.87	0.57	3.887	3.928	9.45	7.137	7.124	6.94	0.55
7	12.07	12.00	9.75	13.06	13.08	6.08	0.92	41.66	41.61	10.7	49.62	49.85	6.98	0.84
8	131.2	131.3	10.9	80.22	80.53	6.15	1.6	493.2	498.6	11.9	348.0	346.9	6.99	1.4
9	1579	1563	12.0	511.6	507.8	6.34	3.1	6316	6296	12.7	2466	2470	7.10	2.6
10	20900	20480	13.2	3640	3629	7.13	5.7	88320	88810	14.0	21590	21620	8.75	4.1

Table 5: Computer times τ_n in seconds for two random phase-space points a and b using a 3.00 GHz Intel Core2 Duo processor for the 4- and 5-dimensional calculation of the n-gluon virtual correction employing the color-ordered and color-dressed method. Corrections have been evaluated twice to check the consistency of the solutions. The n gluons have colors $(ab)_k$ and polarizations κ_k as specified in the text. Also given are the ratios $r_n = \tau_n/\tau_{n-1}$ where τ_n is the time to evaluate the correction for n gluons, in particular $\tau_n = (\tau_n^{(a)} + \tau_n^{(b)})/2$. The τ_n ratios of the ordered versus dressed method are depicted in the respective last columns of the 4- and 5-dimensional case.



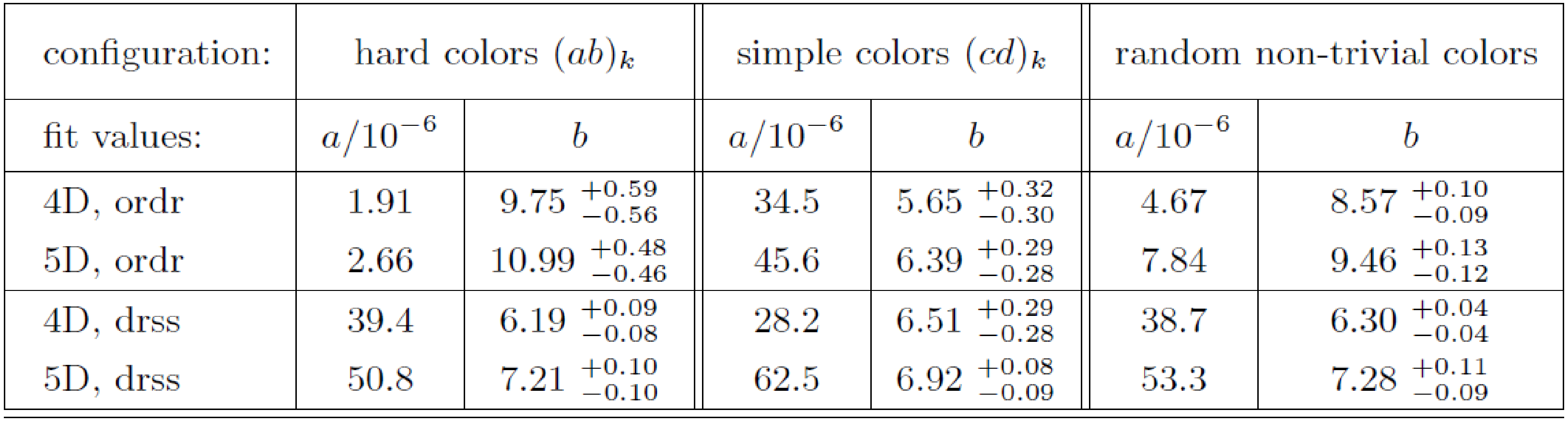
Hard color configuration.

Simple color configuration.

Random color configurations.

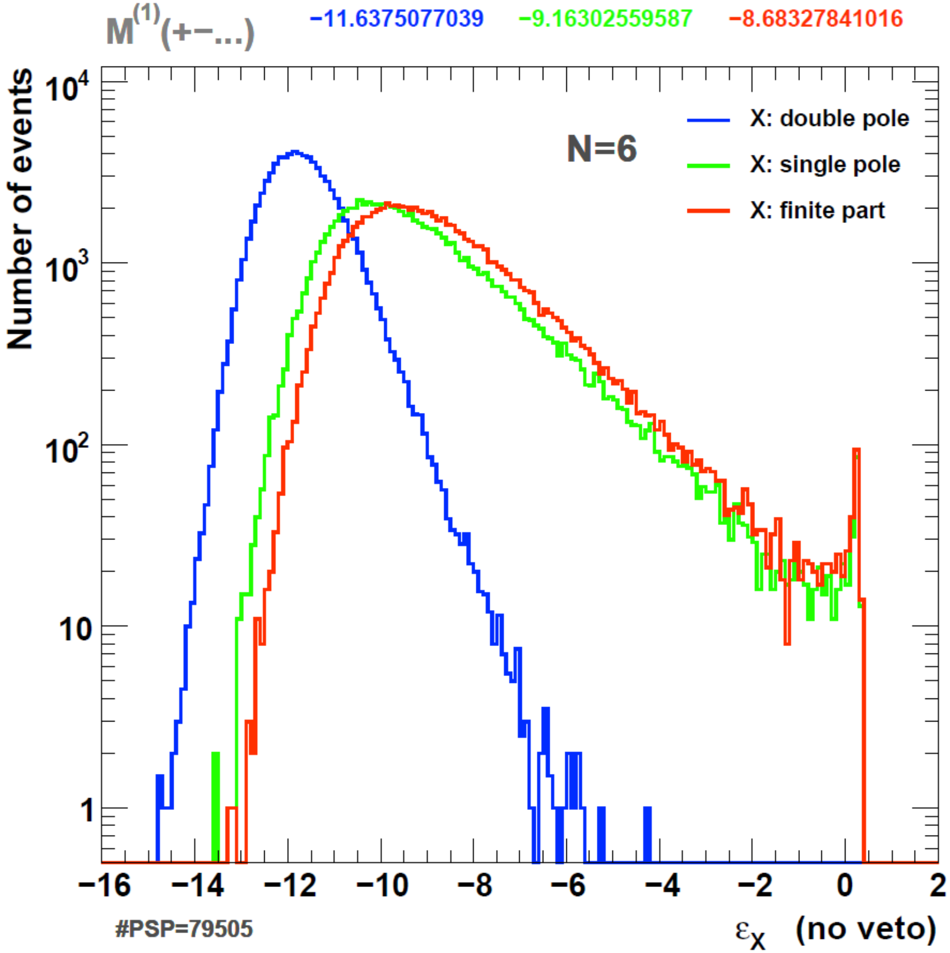
Note that the algorithmic behavior is independent from particle content (e.g. 7-gluon, 7-photon, EW virtual corrections to ggg+t-tbar+b-bbar will all give same order of evaluation time)

$$\tau_n = a * b^n$$



configuration:	hard colors $(ab)_k$		simple colors $(cd)_k$		random non-trivial colors	
fit values:	$a/10^{-6}$	b	$a/10^{-6}$	b	$a/10^{-6}$	b
4D, ordr	1.91	$9.75\ ^{+0.59}_{-0.56}$	34.5	$5.65\ ^{+0.32}_{-0.30}$	4.67	$8.57\ ^{+0.10}_{-0.09}$
5D, ordr	2.66	$10.99\ ^{+0.48}_{-0.46}$	45.6	$6.39\ ^{+0.29}_{-0.28}$	7.84	$9.46\ ^{+0.13}_{-0.12}$
4D, drss	39.4	$6.19\ ^{+0.09}_{-0.08}$	28.2	$6.51\ ^{+0.29}_{-0.28}$	38.7	$6.30\ ^{+0.04}_{-0.04}$
5D, drss	50.8	$7.21\ ^{+0.10}_{-0.10}$	62.5	$6.92\ ^{+0.08}_{-0.09}$	53.3	$7.28\ ^{+0.11}_{-0.09}$

- Apart from timing, accuracy is an important aspect of a numerical implementation
- We are currently focusing on sources of numerical instabilities in the hope to cure (some) of them without resorting the quadrupole (or beyond) precision.
- Just calculating the ***6***-gluon virtual corrections at double precision gives



$$\varepsilon_{\rm dp} \;=\; \log_{10}\frac{|\mathcal{M}^{(1)[1]}_{\rm dp,num}-\mathcal{M}^{(1)}_{\rm dp,th}|}{|\mathcal{M}^{(1)}_{\rm dp,th}|}$$

$$\varepsilon_{\rm s/fp} \;=\; \log_{10}\frac{2\,|\mathcal{M}^{(1)[1]}_{\rm s/fp,num}-\mathcal{M}^{(1)[2]}_{\rm s/fp,num}|}{|\mathcal{M}^{(1)[1]}_{\rm s/fp,num}|+|\mathcal{M}^{(1)[2]}_{\rm s/fp,num}|}$$

Clearly there are numerical instabilities which need to be dealt with.

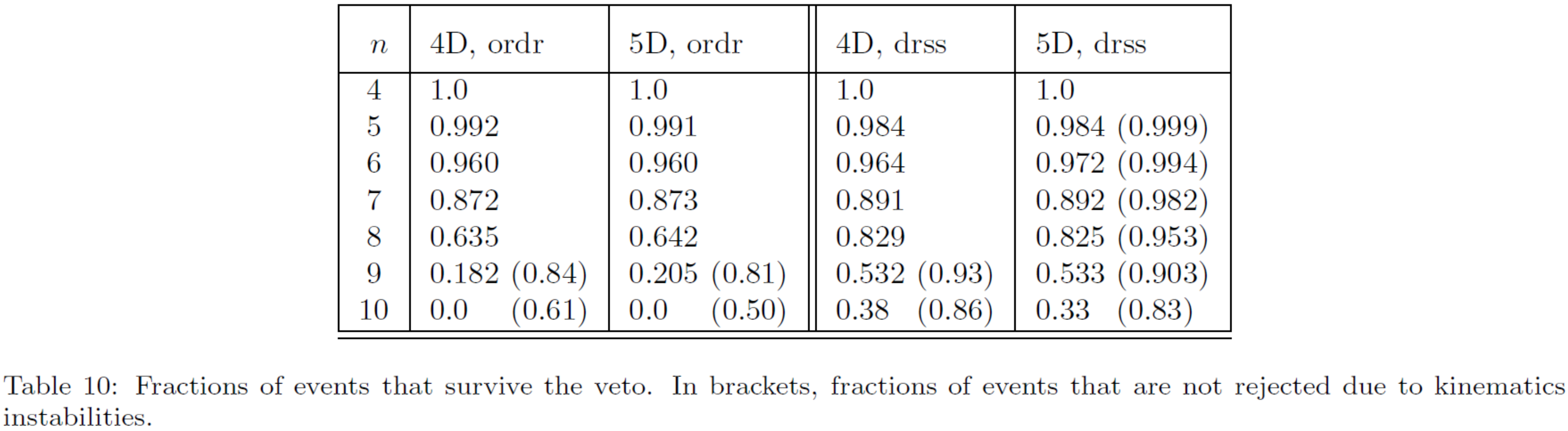
Figure 7: Relative accuracies as obtained from the color-dressed double-precision algorithm for the $1/\epsilon^{2,1,0}$ poles of $n = 6$ gluon one-loop amplitudes with $\lambda_k = +-+-+-$ polarizations and colors chosen randomly among non-trivial configurations. Results are shown without having applied any veto as explained in the text. The mean accuracies and the number of phase-space points are given in the top row and bottom left corner of the plot, respectively.

The main two sources of instabilities are

(1) Given some external momenta, we have to construction additional orthogonal base vectors such that they span the whole 5-dimensional space.
This becomes unstable if the external momenta are nearly linear dependent.
This is the remnant of the so-called “gram-determinant” problem.
However, constructing orthogonal basis is a well classified numerical issue and should be solvable without switching to higher precision.

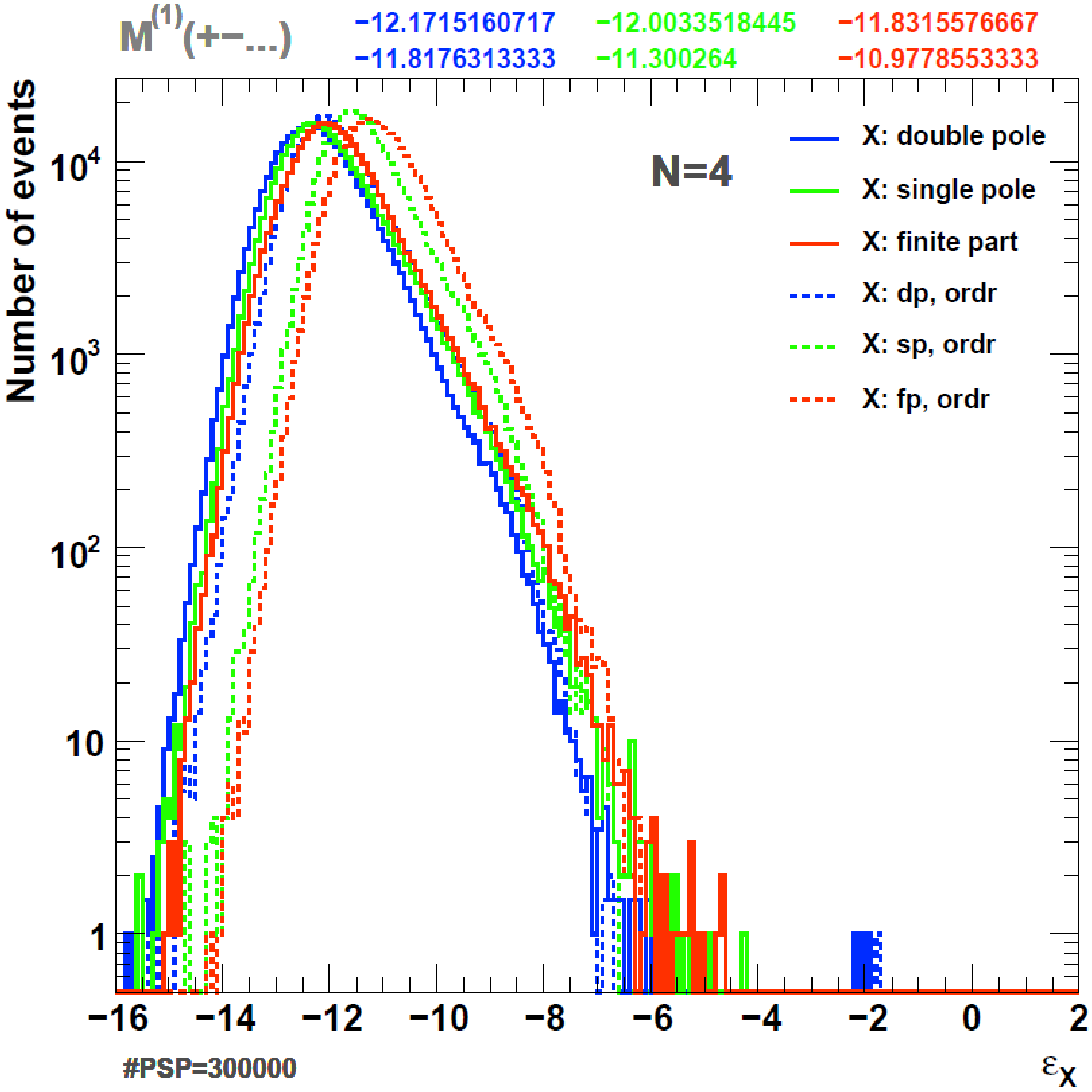
(2) The accumulation of errors into the 4-dimensional part of bubble coefficients (these coefficients depend on subtractions from *all* higher point cuts)
This might be unavoidable within the OPP parametric method, forcing the algorithm to use higher precision (still under investigation)

For now we simply veto events which suffer from these two instabilities.

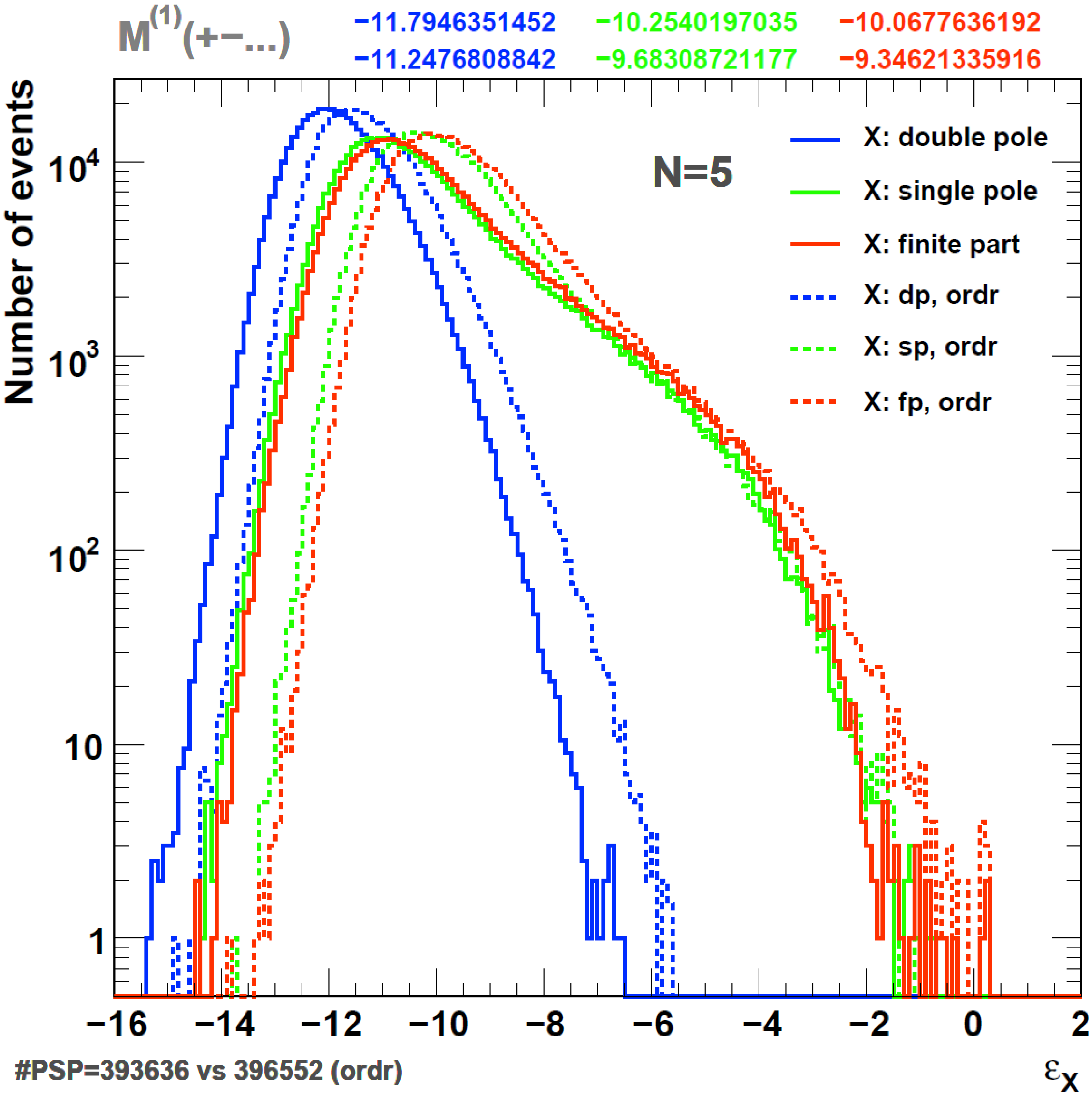


n	4D, ordr	5D, ordr	4D, drss	5D, drss
4	1.0	1.0	1.0	1.0
5	0.992	0.991	0.984	0.984 (0.999)
6	0.960	0.960	0.964	0.972 (0.994)
7	0.872	0.873	0.891	0.892 (0.982)
8	0.635	0.642	0.829	0.825 (0.953)
9	0.182 (0.84)	0.205 (0.81)	0.532 (0.93)	0.533 (0.903)
10	0.0 (0.61)	0.0 (0.50)	0.38 (0.86)	0.33 (0.83)

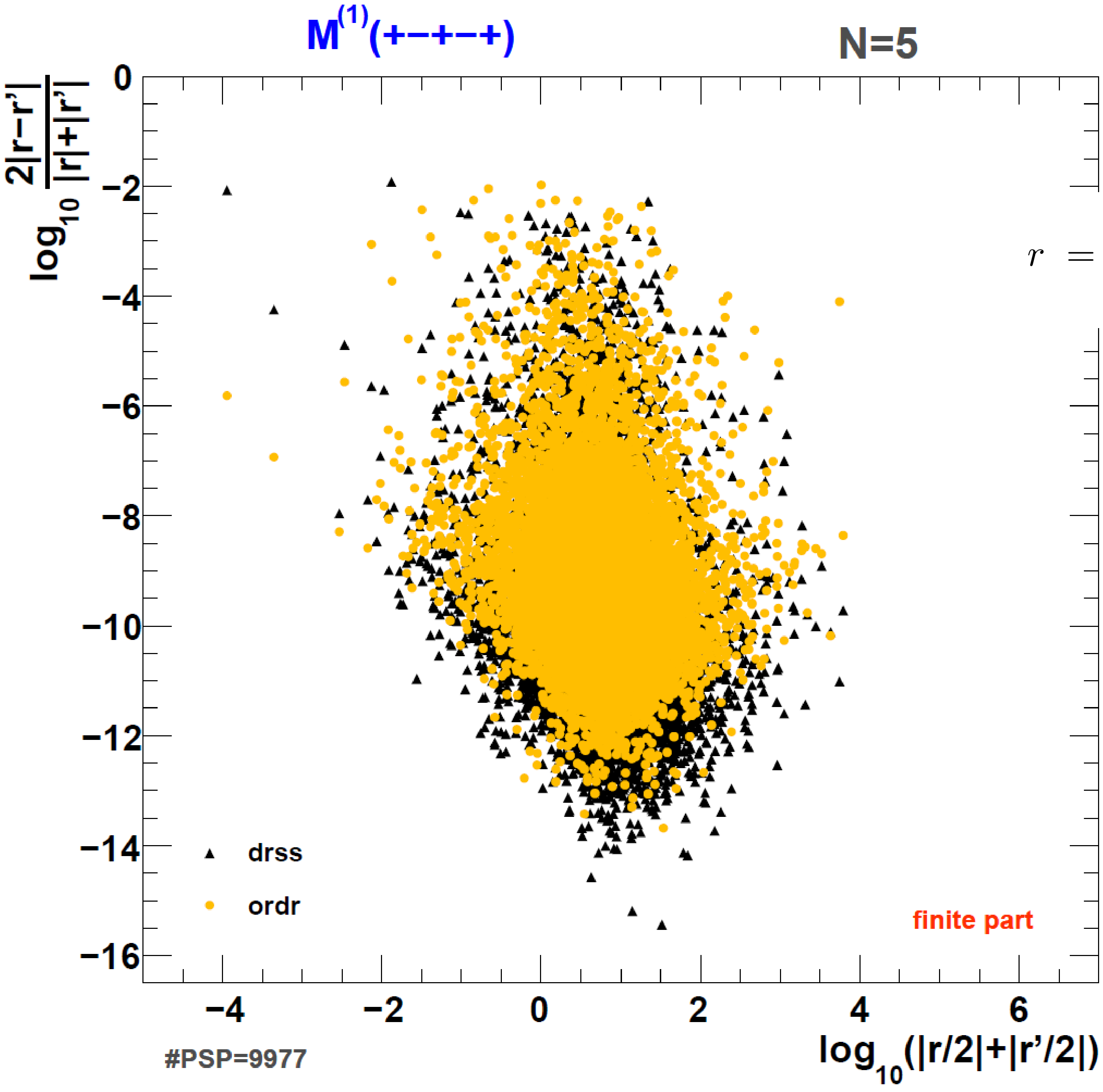
Table 10: Fractions of events that survive the veto. In brackets, fractions of events that are not rejected due to kinematics instabilities.



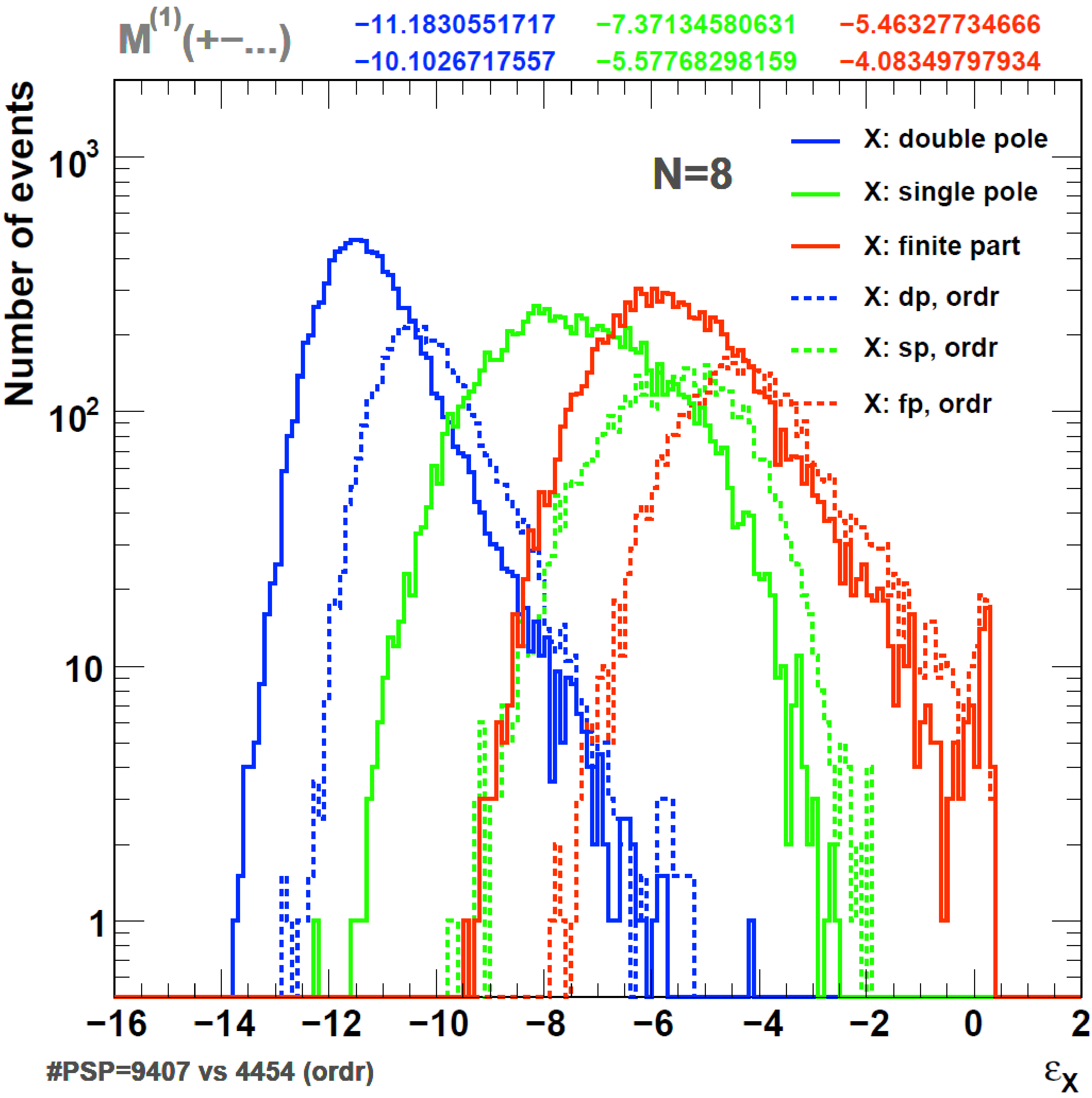
Numerical uncertainties for virtual corrections using dressed and ordered color sampling



It is useful to know the magnitude of the virtual correction given a certain relative uncertainty

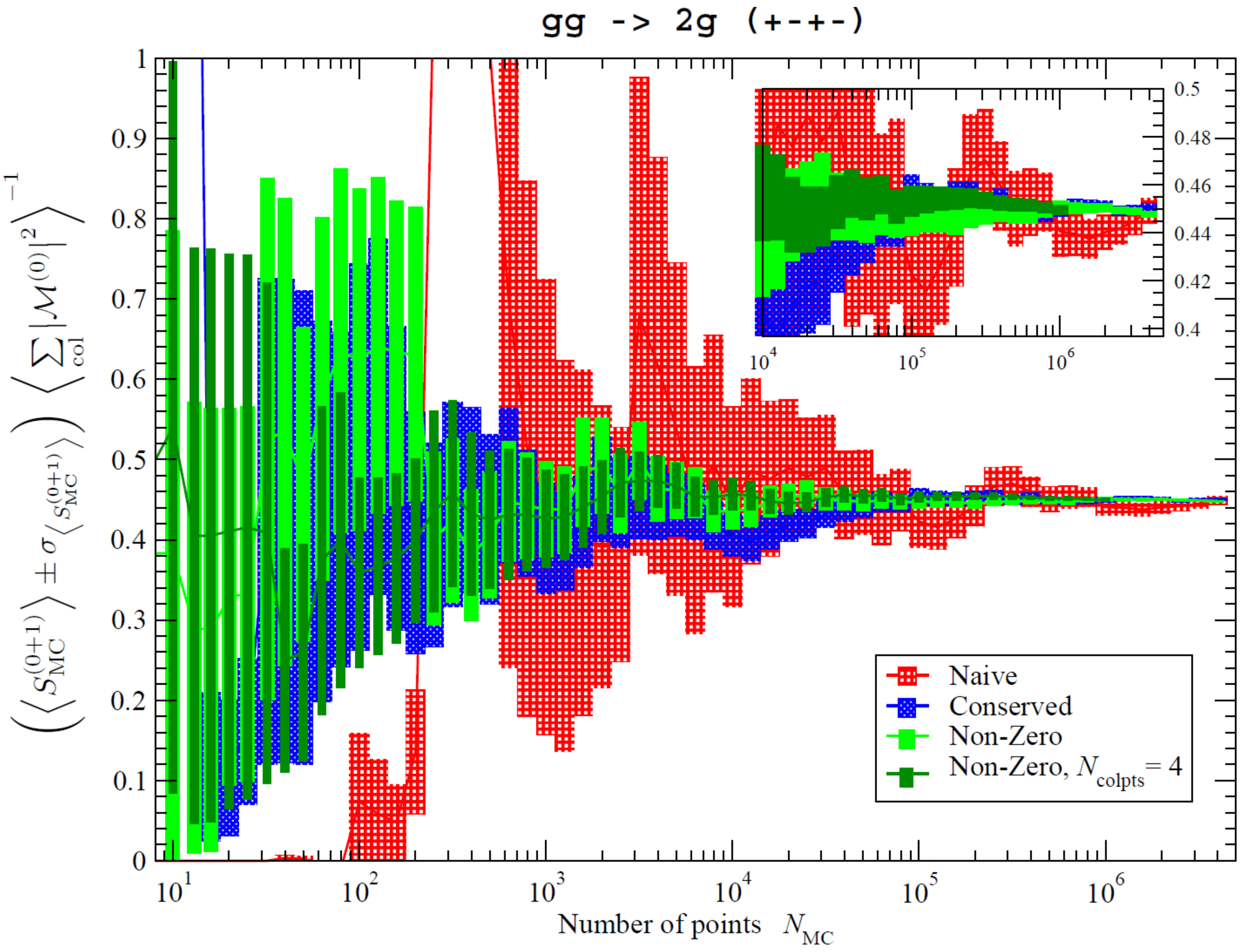


$$r = \frac{2}{16\,\pi^2}\,\frac{\Re\left(\mathcal{M}^{(0)\dagger}\mathcal{M}^{(1)}\right)}{\left|\mathcal{M}^{(0)}\right|^2}$$

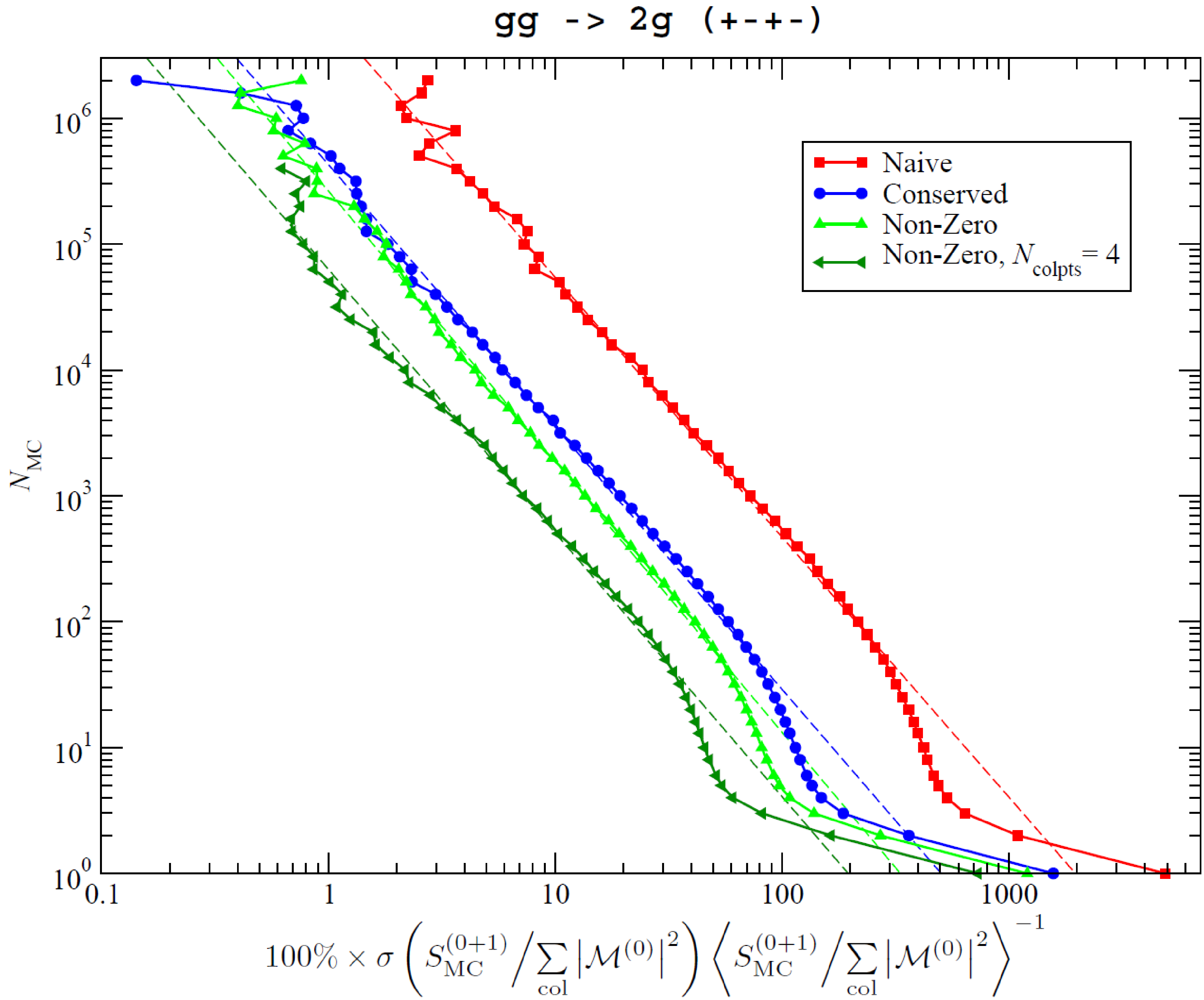


The accuracy difference of ordered vs dressed increases with number of gluons.

The final issue is convergence of phase space integration:



$$S_{\rm MC}^{(0+1)} = \left|\mathcal{M}^{(0)}\right|^2 + \frac{2\hat{\alpha}_S}{4\pi} \Re\left(\mathcal{M}^{(0)\dagger}\mathcal{M}^{(1)}\right)$$


gg -> 2g (+-+-)
Naive
Conserved
Non-Zero
Non-Zero, $N_{\text{colpts}} = 4$
N_{MC}
10^0
10^1
10^2
10^3
10^4
10^5
10^6
0.1
1
10
100
1000
$100\% \times \sigma\left(S_{\text{MC}}^{(0+1)} \Big/ \sum_{\text{col}} \left|\mathcal{M}^{(0)}\right|^2\right) \left\langle S_{\text{MC}}^{(0+1)} \Big/ \sum_{\text{col}} \left|\mathcal{M}^{(0)}\right|^2 \right\rangle^{-1}$

Conclusions:

- We are making good progress on the way to construct a generic NLO generator
- The virtual corrections for arbitrary processes are possible (thanks to particle “blind” dressed recursion relations) with exponential scaling ($\sim 7^n$)
- So far we validated the algorithm by calculating the virtual corrections to multi-gluon scattering
- Validation of the code remains a big issue as many processes become available at once
- There remain speed, accuracy and convergence issues which need to be resolved.
- We have started on adding in real corrections
  - How “automated” are automated subtraction programs?
  - Maybe simpler phase space methods are better (which use brute force phase space point generation)
- This is a large project but we feel confident this can all come together in the coming year(s)
  - First target: NLO QCD corrections to multi-jet processes (+external EW particles).