

## WHIZARD 2.2 for Linear Colliders

W. KILIAN<sup>1a</sup>, F. BACH<sup>2b</sup>, T. OHL<sup>3c</sup>, J. REUTER<sup>4b</sup>

*<sup>a</sup>University of Siegen, Physics Department,  
Walter-Flex-Str. 3, D-57068 Siegen, Germany,*

*<sup>b</sup>DESY Theory Group,  
Notkestr. 85, D-22607 Hamburg, Germany*

*<sup>c</sup>University of Würzburg, Faculty of Physics and Astronomy,  
Am Hubland, D-97074 Würzburg, Germany*

Talk presented at the International Workshop on Future Linear Colliders (LCWS13),  
Tokyo, Japan, 11-15 November 2013

### ABSTRACT

We review the current status of the WHIZARD event generator. We discuss, in particular, recent improvements and features that are relevant for simulating the physics program at a future Linear Collider.

---

<sup>1</sup>kilian@physik.uni-siegen.de

<sup>2</sup>fabian.bach@desy.de

<sup>3</sup>ohl@physik.uni-wuerzburg.de

<sup>4</sup>juergen.reuter@desy.de

# 1 Introduction

WHIZARD [1] is a universal event generator for elementary scattering and decay processes at high-energy colliders.

WHIZARD contains the `O'Mega` matrix element generator [2]. This program generates optimized code for multi-leg tree-level matrix elements in the helicity formalism. The multi-channel Monte-Carlo integrator `VAMP` [3] takes care of high-dimensional integrations. The `VAMP` algorithm is adaptive both between and within channels, and thus computes accurate phase-space integrals and efficiently generates weighted and unweighted event samples.

The WHIZARD core connects these different components. It contains the algorithm for multi-channel phase-space parameterization and mapping, it provides the user interface, interfaces to external programs (e.g., parton distributions, event formats, hadronization), the routines for writing and reading event files, and a parton-shower module. Furthermore, there are optional components for performing numerical analysis and visualization of event samples.

For a realistic description of the ILC and CLIC environments, WHIZARD contains the `CIRCE` [4] package for beam-spectrum simulation. Alternatively, it can digest `GuineaPig` beam-event samples.

## 2 Development

### WHIZARD 1

WHIZARD was initiated in the context of the TESLA design study [5]. It started in the 1990s as a project for an improved calculation of electroweak processes<sup>1</sup> at high-energy lepton colliders, where full multi-leg matrix elements without factorization were required. During the following decade, it was developed towards a universal generator for partonic events at lepton and hadron colliders.

Between 2000 and 2010, important improvements included the implementation of QCD color and thus the full Standard Model (SM), parton distributions, event samples, and support for a growing list of models beyond the SM (BSM). Among the latter, most prominent is the implementation of the full MSSM and further variants of SUSY models such as the NMSSM [6]. WHIZARD supports the SUSY Les Houches Accord (SLHA) [8] for interfacing SUSY spectrum generators. The package has been used for many theory and experimental studies for LHC and TESLA/ILC (see e.g. [7]). In particular, it was used for generating the Linear Collider event database at SLAC.

WHIZARD 1 is no longer actively supported. Development stopped with version 1.94 in 2010. The follow-up releases until 1.97 include only bug and regression fixes and the finalization of the manual.

---

<sup>1</sup>namely, W, Higgs, Z and Respective Decays, thus the acronym

## WHIZARD 2

A thorough rewrite of the core program became necessary for meeting the increased demands regarding applications and versatility of WHIZARD as a tool for phenomenology and experiment. This resulted in the WHIZARD 2 series of releases.

In WHIZARD 2, the static setup of input files has been replaced by a domain-specific language SINDARIN. This user interface allows for simple and compact input in standard use cases, but it also equips the user with the full power of a programming language, adapted to the special needs and conventions of particle physics. It deals with topics such as cuts, analysis, interfaces, process collections, or parameter scans.

WHIZARD 2 make use of OpenMP for parallel execution. New models can be added using the `FeynRules` [9] package [10]. Internally, calculations are performed using a generic density-matrix formalism. Furthermore, the program contains a parton-shower module [11] matched with exact matrix elements via the MLM scheme, as an alternative to external showering.

### The Package

The programming language of WHIZARD 2 is modern Fortran (Fortran 2003), as it is supported by current compilers including the free `gfortran`, version 4.7 and higher. (The subpackage `O'Mega` is written in the functional programming language `OCaml`, which is freely available for all major platforms.) The current WHIZARD production version is 2.1.1 (as of 2012). The complete package works and can be installed on all recent Linux and Mac OS systems. The new version 2.2.0 is currently (March 2014) in beta stage and due to replace the previous version.

## 3 Technical Improvements

### Workflow

The workflow in WHIZARD 2 has been greatly simplified compared to WHIZARD 1, which did involve a conglomerate of shell and Perl scripts, Makefiles and staged compilation.

The installation is possible at a central place, or locally in a user directory. It conforms to the standard toolchain of `automake`, `autoconf`, and `libtool`. The program is maintained at the HepForge server [12] and can be downloaded either as a tagged `.tar.gz` version, or as a development version from the public `svn` repository. Installation of the whole package proceeds via the usual `configure – make – make install` triple and the optional `make check` and `make installcheck` steps for additional peace of mind.

User projects can then be set up in arbitrary directories without any predefined structure. Usually, there is a single command `whizard` and a single SINDARIN input file. Alternative modes of using WHIZARD include an interactive mode, or linking it as a subroutine library that is interoperable with C, C++ or any other C-compatible language (e.g., Python). WHIZARD generates and digests process code on-the-fly in form of dynamically linked libraries. For working with batch clusters, a statically linked mode is also available.

## Internal Structure

WHIZARD 2 is broken down into modules according to the modern Fortran standard. Except for some legacy parts, the program is written in a strict object-oriented fashion. Between versions 2.1 and 2.2, there has been a further major refactoring of the code which introduced an extra abstraction layer for the central types and objects. There is now a consequent separation of interface from implementation, which greatly improves the maintainability and, in particular, the possibilities for future module replacements, reimplementations, and extensions. This affects, for instance, process structure, matrix element calculation methods, beam structure, integration methods, decays and shower.

## Maintenance

The WHIZARD system is maintained with `svn` version control at HepForge. All commits are run automatically through a chain of several hundreds of unit and function tests, using a Jenkins continuous integration server. Bugs and feature requests are handled using HepForge's tracking system.

# 4 Physics

## 0'Mega

The matrix element generator 0'Mega [2] is able to generate exact tree-level matrix elements with multiple external legs (successful tests on standard hardware have involved up to 15 external particles). It is based on a recursive algorithm that reuses *all* common subexpressions and replaces the forest of all tree diagrams by the equivalent directed acyclical graph (DAG). One of the major improvements in version 2 is the treatment of color for QCD amplitudes. It uses the color-flow formalism with phantom  $U(1)$  particles [13] as an efficient way to generate colorized DAGs that can be evaluated exactly or be projected onto color-flow amplitudes.

## Process Definition

WHIZARD now implements the possibility to let processes consist of several different components, which enables one to define process containers for inclusive production samples. This is technically different from flavor sums (also available), where masses have to be identical in order to use the same phase space. In WHIZARD 2.2, this feature will be made available to the user by appropriate syntax in SINDARIN, covering inclusive processes and a detailed specification of decay chains.

## Process Evaluation

The internal density-matrix formalism keeps track of exact spin and color correlations.

Apart from complete matrix elements, a new feature of WHIZARD 2 is the possibility to generate decay chains and cascades. They are composed of an arbitrary choice of elementary

processes which are separately integrated, but concatenated for the event generation. For the intermediate states, the default is to take full spin correlations into account. There is also an option to restrict to classical spin correlations (only diagonal entries in the spin-density matrix), or to even switch off spin correlations completely. If requested, `WHIZARD` will set up decays and branching fractions for the chosen model automatically.

`WHIZARD 2` allows for arbitrary factorization and renormalization scale settings (affecting QCD); the syntax parallels the one for cuts or analysis.

### **Reweighting**

Another new feature within `WHIZARD 2.2` is the possibility to reweight existing event samples, generated either internally or read from file, when changing the setup of the original process, e.g. parameters in the hard matrix elements, the event scale, the chosen structure functions, or the QCD parton shower.

## **5 Linear Collider Simulation**

`WHIZARD` has been developed as a generator for any high-energy collider, but it contains modules that are dedicated to the description of a lepton collider environment. In particular, given the required level of precision at ILC and CLIC, an accurate description of beam properties is essential for studies and analyses.

### **Beamstrahlung**

The program uses an interface to the `CIRCE1` [4] program that parameterizes the beams of an  $e^+e^-$  collider and provides an event generator for factorized beam spectra. Interfacing this generator (or, alternatively, the parameterized spectrum directly), `WHIZARD` can integrate and simulate any  $e^+e^-$  process with a realistic beam description. As an upgrade to previous versions, `WHIZARD 2.2` ships with beam spectra that correspond to the current ILC design parameters. To account for cases where such a factorized form is insufficient, `WHIZARD` can alternatively read beam-event files as they are produced by `GuineaPig`.

### **ISR**

On top of the beamstrahlung effects, lepton-collider processes are strongly affected by electromagnetic initial-state radiation (ISR). `WHIZARD` implements ISR in a standard structure-function formalism that resums the corrections from infrared (leading) and collinear (3rd order) radiation and implements them in kinematics and dynamics, if requested.

### **Polarization**

Furthermore, `WHIZARD` allows for specifying beam polarization, ranging from unpolarized, left- or right-handed circular or transversal polarization to arbitrary spin-density matrices. The polarization and polarization fractions are specified for both beams independently. The user

can also define asymmetric beam setups and a crossing angle, which will be taken into account in the kinematics setup.

## Photons

Photons as initial particles are available in various incarnations: on-shell, radiated from  $e^\pm$  (effective photon approximation), or beamstrahlung photons generated by `CIRCE1`. A photon-collider option that uses the `CIRCE2` beam description is also available but no longer maintained, due to the lack of current ILC or CLIC photon-collider mode beam parameters and simulations of the corresponding beam-beam interactions.

## Events

For cuts, reweighting and internal analysis, `WHIZARD` employs its dedicated language `SINDARIN` that allows for computing a wide range of event-specific and generic observables.

`WHIZARD` supports various event output formats, including the traditional `HEPEVT` format and its derivatives, `StdHEP`, `LHEF`, `HepMC`, and others. A direct interface to `LCIO` is planned.

# 6 QCD

## Parton Shower

For a precise calculation for exclusive Linear Collider processes, a program needs fine control over QCD corrections. Regarding real radiation, the multi-leg capability of `WHIZARD` allows for including high orders of the QCD coupling. Collinear and soft radiation in exclusive events is affected by large logarithms, which are conveniently resummed in the semi-classical approach of a parton shower algorithm.

Using standard event formats and suitable cuts, `WHIZARD` allows for attaching an external parton-shower generator. `WHIZARD 2` furthermore contains an internal showering module in two different incarnations: a  $k_T$  ordered shower along the lines of the `Pythia` shower [14], and an analytic parton shower, which keeps the complete shower history and allows to reweight it [11]. There is support for combining exact matrix elements and QCD radiation from the parton shower using the MLM matching prescription. These modules are forseen to receive a more detailed validation, tuning and further improvements after the 2.2 release.

Beyond the (partonic) parton shower, hadronization and hadronic decays are not performed by `WHIZARD` internally, but can be applied to the generated partonic event samples via the `Pythia 6` package [14] which is attached to the `WHIZARD` distribution, or using e.g. `LHE` event samples that are then fed into a different external (shower and) hadronization package.

## Virtual Corrections and Subtraction

`WHIZARD 2` with `0'Mega` matrix elements is an event generator of tree-level processes. There have been several projects that extended it to next-to-leading order, including loop corrections and proper infrared-collinear subtraction. Ref. [15,16] describes the extension of `WHIZARD 1`

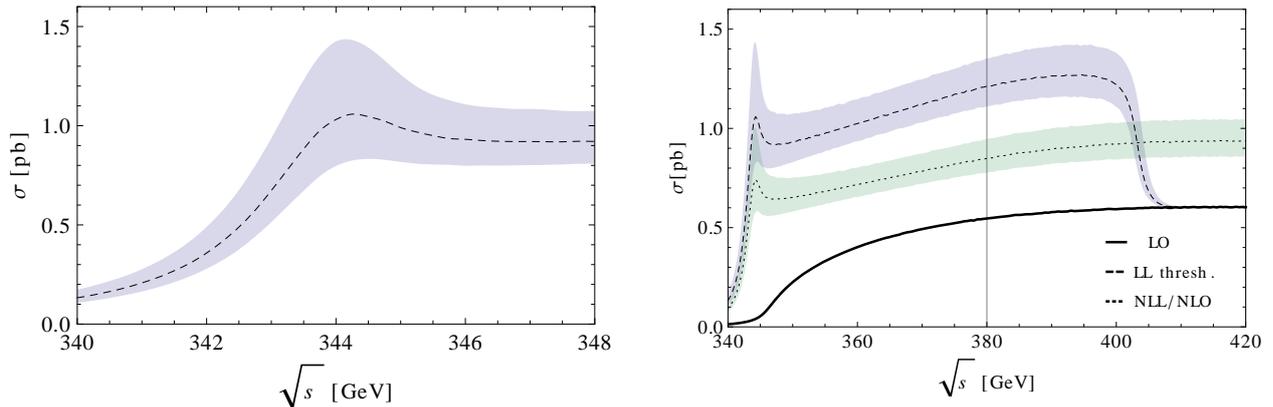


Figure 1: *WHIZARD* total  $e^+e^- \rightarrow t\bar{t} \rightarrow b\bar{b}W^+W^-$  cross section including non-relativistic threshold corrections at LL order: threshold region (left) reproducing the shape e. g. in [22], and matching to continuum (right), which is at the current status of progress achieved by adapting  $K$ -factors for NLL threshold and NLO continuum normalization (uncertainties from variation of the soft renormalization scale).

to a positive-definite NLO event generator for the electroweak pair production of charginos in the MSSM, including full electroweak SUSY corrections matched to the photon initial and final state radiation. Independently, Ref. [17,18] implemented the QCD NLO correction with subtraction for a particular LHC process. Along these lines, the Binoth Les Houches Accord (BLHA) interface [19] has been implemented for reading and writing contract files with one-loop programs (OLP), and has been validated.

Building upon the new data structures of *WHIZARD* 2.2, an implementation of automatic NLO QCD corrections is currently being developed.

### Top-Quark Threshold

A high-luminosity linear collider will be capable of a high-precision scan of the top-quark pair-production threshold [20]. To match this on the theoretical side, one needs to resum logarithms of the top velocity  $\sim \alpha_s \ln v$  as well as gluon Coulomb potential terms  $\sim \alpha_s/v$  in a non-relativistic approach and to relate this to the relativistic matrix elements in the continuum. There is an ongoing project for including these effects in *WHIZARD* 2 which will make the theoretical calculation available in the simulation of exclusive events. As a first step, the leading-logarithmic approximation matched to leading-order matrix elements using a relative  $K$ -factor has already been implemented, cf. Fig. 1, and will be included in an upcoming release [21].

## 7 Physics models

### Particles

As a generator of hard matrix elements, **WHIZARD** has to support various particle species and interactions. The allowed spin representations for particles are 0, 1, 2 (bosons) and 1/2, 3/2 (fermions), all massive or massless, both Dirac and Majorana spinors, optionally colored (triplet or octet). The **WHIZARD** libraries support all Lorentz structures for interactions in the models described below. A completely general framework supporting all possible Lorentz structures is under construction.

### BSM Models

Beyond the SM and its QCD and QED subsets, **WHIZARD** supports the minimal supersymmetric Standard Model (MSSM) with different variants and extensions. These include models with gravitinos [7] and the NMSSM [6] (see also [23]).

Among models with strongly interacting sectors **WHIZARD** includes Little Higgs models in different incarnations, with and without discrete symmetries, cf. [24].

**WHIZARD** has also been used for studying more exotic models such as the noncommutative SM [25] (not included in the official release). It further supports the completely general two-Higgs doublet model (2HDM), as well as generic models containing a  $Z'$  state, and extra-dimensional models like Universal Extra Dimensions (UED). A more detailed list can be found in the **WHIZARD** documentation [12].

### Effective Theories

As an alternative tool for studying deviations from the SM, **WHIZARD** contains SM extensions with anomalous couplings, expressible as coefficients of higher-dimensional operators in an effective theory. Several models in **WHIZARD**'s library define either anomalous triple and quartic gauge boson couplings, which have been used for studies at LCs [26] or at LHC [27,28,29]. Anomalous top-quark couplings are also supported [30].

Recent interest in the physics of high-energy vector-boson scattering has triggered the development and addition of simplified models for strong interactions and compositeness (SSC) [31]. In addition to generic new degrees of freedom, they implement a unitarization procedure that is required for extrapolating into the energy range that will become accessible at ILC and, in particular, at CLIC.

## 8 Conclusion and Outlook

**WHIZARD** is a versatile and user-friendly tool for both SM and BSM physics at all kinds of high-energy colliders. Special emphasis has always been put on the particular requirements for an accurate simulation of exclusive events in the linear-collider environment. This involves a detailed account of the nontrivial beam properties, and of polarized and non-collinear beam configurations.

The WHIZARD collaboration actively pursues the further development of the package, in particular regarding the physics program at ILC and CLIC. This required a thorough refactoring for version 2.2, which subsequently will be the base for further development. Plans for new features include the support for more general Lorentz and color structures in models, a more convenient model interface, power-counting of coupling constants in the matrix element, further refinements in the beam description, and automatic support for higher-order corrections.

## Acknowledgments

JRR has been partially supported by the Strategic Alliance for Terascale Physics of the Helmholtz-Gemeinschaft. TO is supported by the German Ministry of Education and Research (BMBF). WK and JRR want to thank the organizers for scheduling the conference in the amazing Japanese autumn foliage season.

## References

- [1] W. Kilian, T. Ohl and J. Reuter, *Eur. Phys. J. C* **71**, 1742 (2011) [arXiv:0708.4233 [hep-ph]];
- [2] M. Moretti, T. Ohl and J. Reuter, In \*2nd ECFA/DESY Study 1998-2001\* 1981-2009 [hep-ph/0102195]; J. Reuter, hep-th/0212154; T. Ohl, *AIP Conf. Proc.* **583**, 173 (2001) [hep-ph/0011243].
- [3] T. Ohl, *Comput. Phys. Commun.* **120**, 13 (1999) [hep-ph/9806432].
- [4] T. Ohl, *Comput. Phys. Commun.* **101**, 269 (1997) [hep-ph/9607454].
- [5] W. Kilian, In \*2nd ECFA/DESY Study 1998-2001\* 1924-1980
- [6] J. Reuter and F. Braam, *AIP Conf. Proc.* **1200**, 470 (2010) [arXiv:0909.3059 [hep-ph]].
- [7] N. Pietsch, J. Reuter, K. Sakurai and D. Wiesler, *JHEP* **1207**, 148 (2012) [arXiv:1206.2146 [hep-ph]]; J. Kalinowski, W. Kilian, J. Reuter, T. Robens and K. Rolbiecki, *JHEP* **0810**, 090 (2008) [arXiv:0809.3997 [hep-ph]]; K. Hagiwara, W. Kilian, F. Krauss, T. Ohl, T. Plehn, D. Rainwater, J. Reuter and S. Schumann, *Phys. Rev. D* **73**, 055005 (2006) [hep-ph/0512260].
- [8] B. C. Allanach, M. Battaglia, G. A. Blair, M. S. Carena, A. De Roeck, A. Dedes, A. Djouadi and D. Gerdes *et al.*, *Eur. Phys. J. C* **25**, 113 (2002) [hep-ph/0202233]; B. C. Allanach, C. Balazs, G. Belanger, M. Bernhardt, F. Boudjema, D. Choudhury, K. Desch and U. Ellwanger *et al.*, *Comput. Phys. Commun.* **180**, 8 (2009) [arXiv:0801.0045 [hep-ph]]; J. A. Aguilar-Saavedra, A. Ali, B. C. Allanach, R. L. Arnowitt, H. A. Baer, J. A. Bagger, C. Balazs and V. D. Barger *et al.*, *Eur. Phys. J. C* **46**, 43 (2006) [hep-ph/0511344].

- [9] N. D. Christensen and C. Duhr, *Comput. Phys. Commun.* **180**, 1614 (2009) [arXiv:0806.4194 [hep-ph]].
- [10] N. D. Christensen, C. Duhr, B. Fuks, J. Reuter and C. Speckner, *Eur. Phys. J. C* **72**, 1990 (2012) [arXiv:1010.3251 [hep-ph]].
- [11] W. Kilian, J. Reuter, S. Schmidt and D. Wiesler, *JHEP* **1204**, 013 (2012) [arXiv:1112.1039 [hep-ph]].
- [12] <http://whizard.hepforge.org/>
- [13] W. Kilian, T. Ohl, J. Reuter and C. Speckner, *JHEP* **1210**, 022 (2012) [arXiv:1206.3700 [hep-ph]].
- [14] T. Sjostrand, S. Mrenna and P. Z. Skands, *JHEP* **0605**, 026 (2006) [hep-ph/0603175].
- [15] T. Robens, J. Kalinowski, K. Rolbiecki, W. Kilian and J. Reuter, *Acta Phys. Polon. B* **39**, 1705 (2008) [arXiv:0803.4161 [hep-ph]].
- [16] W. Kilian, J. Reuter and T. Robens, *Eur. Phys. J. C* **48**, 389 (2006) [hep-ph/0607127].
- [17] T. Binoth, N. Greiner, A. Guffanti, J. Reuter, J. P. Guillet and T. Reiter, *Phys. Lett. B* **685**, 293 (2010) [arXiv:0910.4379 [hep-ph]].
- [18] N. Greiner, A. Guffanti, T. Reiter and J. Reuter, *Phys. Rev. Lett.* **107**, 102002 (2011) [arXiv:1105.3624 [hep-ph]].
- [19] T. Binoth, F. Boudjema, G. Dissertori, A. Lazopoulos, A. Denner, S. Dittmaier, R. Frederix and N. Greiner *et al.*, *Comput. Phys. Commun.* **181**, 1612 (2010) [arXiv:1001.1307 [hep-ph]]; S. Alioli, S. Badger, J. Bellm, B. Biedermann, F. Boudjema, G. Cullen, A. Denner and H. Van Deurzen *et al.*, *Comput. Phys. Commun.* **185**, 560 (2014) [arXiv:1308.3462 [hep-ph]].
- [20] H. Baer, T. Barklow, K. Fujii, Y. Gao, A. Hoang, S. Kanemura, J. List and H. E. Logan *et al.*, arXiv:1306.6352 [hep-ph].
- [21] F. Bach, W. Kilian, J. Reuter, M. Stahlhofen, in preparation.
- [22] A. Hoang and M. Stahlhofen, [arXiv:1309.6323 [hep-ph]].
- [23] J. Reuter and D. Wiesler, *Phys. Rev. D* **84**, 015012 (2011) [arXiv:1010.4215 [hep-ph]].
- [24] J. Reuter, M. Tonini and M. de Vries, *JHEP* **1402**, 053 (2014) [arXiv:1310.2918 [hep-ph]]; J. Reuter and M. Tonini, *JHEP* **1302**, 077 (2013) [arXiv:1212.5930 [hep-ph]]; W. Kilian, D. Rainwater and J. Reuter, *Phys. Rev. D* **74**, 095003 (2006) [Erratum-ibid. *D* **74**, 099905 (2006)] [hep-ph/0609119]; *Phys. Rev. D* **71**, 015008 (2005) [hep-ph/0411213]; W. Kilian and J. Reuter, *Phys. Rev. D* **70**, 015004 (2004) [hep-ph/0311095].

- [25] A. Alboteanu, T. Ohl and R. Rückl, eConf C **0705302**, TEV05 (2007) [Acta Phys. Polon. B **38**, 3647 (2007)] [arXiv:0709.2359 [hep-ph]]; Phys. Rev. D **74**, 096004 (2006) [hep-ph/0608155]; T. Ohl and J. Reuter, Phys. Rev. D **70**, 076007 (2004) [hep-ph/0406098].
- [26] M. Beyer, W. Kilian, P. Krstonošić, K. Mönig, J. Reuter, E. Schmidt and H. Schröder, Eur. Phys. J. C **48**, 353 (2006) [hep-ph/0604048].
- [27] E. Boos, H. J. He, W. Kilian, A. Pukhov, C. P. Yuan and P. M. Zerwas, Phys. Rev. D **57**, 1553 (1998) [hep-ph/9708310].
- [28] A. Alboteanu, W. Kilian and J. Reuter, JHEP **0811**, 010 (2008) [arXiv:0806.4145 [hep-ph]].
- [29] J. Reuter, W. Kilian and M. Sekulla, arXiv:1307.8170 [hep-ph].
- [30] F. Bach and T. Ohl, Phys. Rev. D **86**, 114026 (2012) [arXiv:1209.4564 [hep-ph]]; F. Bach and T. Ohl, in preparation.
- [31] J. Reuter, W. Kilian, M. Sekulla, DESY 14-044, SI-HEP-2014-06, these proceedings; W. Kilian, J. Reuter, M. Sekulla, in preparation.