

// WARNING: Needs to be run with ROOT 6.14 or newer! (Else creating .C files is a meeeeeess)

```
float string_to_float( string str ) {  
    return strtod((str).c_str(),0);  
}
```

```
shared_ptr get_comparison_stack( string comparison_name, TH1D* level1_m_WW, TH1D*  
level1_m_ZZ, TH1D* level2_m_WW, TH1D* level2_m_ZZ, TH1D* level3_m_WW = NULL,  
TH1D* level3_m_ZZ = NULL) {  
    level1_m_WW->SetLineColor(9124);  
    level1_m_ZZ->SetLineColor(9123);  
    level2_m_WW->SetLineColor(9024);  
    level2_m_ZZ->SetLineColor(9023);
```

```
    level1_m_WW->SetLineStyle(9);  
    level1_m_ZZ->SetLineStyle(9);  
    level2_m_WW->SetLineStyle(2);  
    level2_m_ZZ->SetLineStyle(2);
```

```
    level1_m_WW->SetLineWidth(3);  
    level1_m_ZZ->SetLineWidth(3);  
    level2_m_WW->SetLineWidth(3);  
    level2_m_ZZ->SetLineWidth(3);
```

```
    shared_ptr comparison_stack = make_shared( comparison_name.c_str(), "(m_{jj},1) +  
m_{jj,2})/2 [GeV]; Events" );  
    comparison_stack->Add(level1_m_WW);  
    comparison_stack->Add(level1_m_ZZ);  
    comparison_stack->Add(level2_m_WW);  
    comparison_stack->Add(level2_m_ZZ);
```

```
    if ( NULL != level3_m_WW && NULL != level3_m_ZZ ) {  
        level3_m_WW->SetLineColor(kBlue); //TODO Adjust  
        level3_m_ZZ->SetLineColor(kRed); //TODO Adjust  
        level3_m_WW->SetLineStyle(1); // TODO Adjust  
        level3_m_ZZ->SetLineStyle(1); // TODO Adjust  
        level3_m_WW->SetLineWidth(3);  
        level3_m_ZZ->SetLineWidth(3);  
        comparison_stack->Add(level3_m_WW);  
        comparison_stack->Add(level3_m_ZZ);  
    }
```

```

return comparison_stack;
}

```

```

TH1D* get_ratio_plot( string ratio_name, TH1D* numerator, TH1D* denominator) {
TH1D* ratio_plot = (TH1D*)numerator->Clone(ratio_name.c_str());
ratio_plot->Sumw2();
ratio_plot->Divide(denominator);
ratio_plot->SetTitle(""); // Remove the ratio title
return ratio_plot;
}

```

```

shared_ptr get_ratio_stack( string ratio_name, TH1D* numerator_m_WW, TH1D*
numerator_m_ZZ, TH1D* denominator_m_WW, TH1D* denominator_m_ZZ, string y_title ) {
shared_ptr ratio_stack = make_shared( ratio_name.c_str(), ("(m_{jj,1} + m_{jj,2})/2 [GeV];" +
y_title).c_str() );
TH1D* ratio_WW = get_ratio_plot( (ratio_name+"_WW").c_str(), numerator_m_WW,
denominator_m_WW );
TH1D* ratio_ZZ = get_ratio_plot( (ratio_name+"_ZZ").c_str(), numerator_m_ZZ,
denominator_m_ZZ );
ratio_WW->SetLineColor(kBlue);
ratio_ZZ->SetLineColor(kRed);
ratio_WW->SetFillColorAlpha(kBlue, 0.4);
ratio_ZZ->SetFillColorAlpha(kRed, 0.4);
ratio_WW->SetLineStyle(1);
ratio_ZZ->SetLineStyle(1);
ratio_WW->SetMarkerColor(kBlue);
ratio_ZZ->SetMarkerColor(kRed);
ratio_WW->SetMarkerStyle(23);
ratio_ZZ->SetMarkerStyle(23);
ratio_stack->Add(ratio_ZZ);
ratio_stack->Add(ratio_WW);
return ratio_stack;
}

```

```

void renormalize_axis_sizes( shared_ptr stack, double canvas_size_ratio ) {
/* When using different size pads the sizes are scaled with pad size so one of
them will be off -> renormalize them again!
/ auto x_axis = stack->GetHistogram()->GetXaxis(); auto y_axis = stack->GetHistogram()-
>GetYaxis(); y_axis->SetTitleSize(y_axis->GetTitleSize()*canvas_size_ratio);
y_axis->SetTitleOffset(y_axis->GetTitleOffset()/canvas_size_ratio);
y_axis->SetLabelSize(y_axis->GetLabelSize()*canvas_size_ratio); x_axis->SetTitleSize(x_axis-
>GetTitleSize()*canvas_size_ratio);
x_axis->SetLabelSize(x_axis->GetLabelSize()*canvas_size_ratio);

```

```
}
```

```
shared_ptr add_ILD_mark( shared_ptr canvas, Double_t x0, Double_t y0, Float_t text_size=0.25 )  
{  
    canvas->cd();  
    shared_ptr ild_tex (new TLatex(x0,y0,"ILD "));  
    ild_tex->SetName( (string(canvas->GetName()) + "_ildlogo").c_str() );  
    ild_tex->SetTextSize(text_size);  
    ild_tex->SetTextFont(62);  
    ild_tex->SetLineWidth(2);  
    ild_tex->Draw();  
    return ild_tex;  
}
```

```
shared_ptr add_prelim_mark( shared_ptr canvas, Double_t x0, Double_t y0, Float_t text_size=0.2  
) {  
    canvas->cd();  
    shared_ptr prelim_tex (new TLatex(x0,y0,"preliminary"));  
    prelim_tex->SetName( (string(canvas->GetName()) + "_prelim_text").c_str() );  
    prelim_tex->SetTextSize(text_size);  
    prelim_tex->SetTextFont(42);  
    prelim_tex->SetLineWidth(2);  
    prelim_tex->Draw();  
    return prelim_tex;  
}
```

```
shared_ptr mark_preliminary( shared_ptr canvas, Double_t x0, Double_t y0, Float_t  
text_size=0.25, Float_t text_angle=60 ) {  
    canvas->cd();  
    cout << gPad->GetName() << endl; shared_ptr prelim_tex (new TLatex(x0,y0,"Preliminary"));  
    prelim_tex->SetName( (string(canvas->GetName()) + "_prelim").c_str() );  
    prelim_tex->SetTextColorAlpha(17, 0.5);  
    prelim_tex->SetTextSize(text_size);  
    prelim_tex->SetTextAngle(text_angle);  
    prelim_tex->SetLineWidth(2);  
    prelim_tex->Draw();  
    return prelim_tex;  
}
```

```
void adjust_canvas_horizontal_to_square_pad(shared_ptr canvas) {  
    double vertical_square_fraction = 1.0 - canvas->GetTopMargin() - canvas->GetBottomMargin();  
    double width_to_height_ratio = double(canvas->GetWindowWidth()) / double(canvas->GetWindowHeight());
```

```

double horizontal_square_fraction = vertical_square_fraction / width_to_height_ratio;
double left_margin = 1.0 - canvas->GetRightMargin() - horizontal_square_fraction;
cout << left_margin << " " << canvas->GetWindowWidth() << " " << canvas->GetWindowHeight() << std::endl; canvas->SetLeftMargin(left_margin);
}

void create_detector_and_icn_VV_m_comparison() {
// Standard ROOT plot setting
gROOT->Reset();
gStyle->SetOptStat(0);
TH2::SetDefaultSumw2();

string output_dir =
"/afs/desy.de/group/flc/pool/beyerjac/VBS/nunu_hadron/comparisons/comparisons_ls_and_cheated/";

string l5_directory = "/afs/desy.de/group/flc/pool/beyerjac/VBS/nunu_hadron/v02-00-02_l5_o1_v02_output";
string s5_directory = "/afs/desy.de/group/flc/pool/beyerjac/VBS/nunu_hadron/v02-00-02_s5_o1_v02_output";

shared_ptr file_l5 = make_shared( (l5_directory + "/mjjmjj_plots/_all_TH1Ds.root").c_str() );
shared_ptr file_l5_2D = make_shared( (l5_directory + "/mjjmjj_plots/_all_TH2Ds.root").c_str() );
shared_ptr file_l5_icn = make_shared( (l5_directory +
"/TJ/TJ_observ_to_icns/_all_TH1Ds.root").c_str() );
shared_ptr file_l5_icn_2D = make_shared( (l5_directory +
"/TJ/TJ_observ_to_icns/_all_TH2Ds.root").c_str() );
shared_ptr file_l5_icn_udsonly = make_shared( (l5_directory +
"/TJ/TJ_observ_to_icns_uds_only/_all_TH1Ds.root").c_str() );
shared_ptr file_s5 = make_shared( (s5_directory + "/mjjmjj_plots/_all_TH1Ds.root").c_str() );
shared_ptr file_s5_2D = make_shared( (s5_directory + "/mjjmjj_plots/_all_TH2Ds.root").c_str() );
shared_ptr file_s5_icn = make_shared( (s5_directory +
"/TJ/TJ_observ_to_icns/_all_TH1Ds.root").c_str() );
shared_ptr file_s5_icn_2D = make_shared( (s5_directory +
"/TJ/TJ_observ_to_icns/_all_TH2Ds.root").c_str() );
shared_ptr file_s5_icn_udsonly = make_shared( (s5_directory +
"/TJ/TJ_observ_to_icns_uds_only/_all_TH1Ds.root").c_str() );

shared_ptr file_l5_SLDs = make_shared( (l5_directory +
"/mjj_vs_SLDdecays/_all_TH1Ds.root").c_str() );
shared_ptr file_s5_SLDs = make_shared( (s5_directory +
"/mjj_vs_SLDdecays/_all_TH1Ds.root").c_str() );
vector< closables {file_l5, file_l5_icn, file_l5_icn_udsonly, file_s5, file_s5_icn,

```

file_s5_icn_udsonly, file_I5_SLDs, file_s5_SLDs};

TH1D *m_WW_genlevel* = (TH1D)file_I5->Get("m_WW_genlevel") ;

TH1D *m_ZZ_genlevel* = (TH1D)file_I5->Get("m_ZZ_genlevel") ;

TH1D *m_WW_I5* = (TH1D)file_I5->Get("m_WW_nocuts") ;

TH2D *m_m_WW_I5* = (TH2D)file_I5_2D->Get("m_m_WW_nocuts") ;

TH1D *m_WW_I5_icn* = (TH1D)file_I5_icn->Get("m_WW_nocuts") ;

TH1D *m_WW_I5_cheatcluster* = (TH1D)file_I5_icn->Get("m_WW_custom_pairing_nocuts") ;

TH1D *m_WW_I5_icn_udsonly* = (TH1D)file_I5_icn_udsonly->Get("m_WW_uds_nocuts") ;

TH1D *m_ZZ_I5* = (TH1D)file_I5->Get("m_ZZ_nocuts") ;

TH2D *m_m_ZZ_I5* = (TH2D)file_I5_2D->Get("m_m_ZZ_nocuts") ;

TH1D *m_ZZ_I5_icn* = (TH1D)file_I5_icn->Get("m_ZZ_nocuts") ;

TH1D *m_ZZ_I5_cheatcluster* = (TH1D)file_I5_icn->Get("m_ZZ_custom_pairing_nocuts") ;

TH1D *m_ZZ_I5_icn_udsonly* = (TH1D)file_I5_icn_udsonly->Get("m_ZZ_uds_nocuts") ;

TH1D *m_WW_s5* = (TH1D)file_s5->Get("m_WW_nocuts") ;

TH2D *m_m_WW_s5* = (TH2D)file_s5_2D->Get("m_m_WW_nocuts") ;

TH1D *m_WW_s5_icn* = (TH1D)file_s5_icn->Get("m_WW_nocuts") ;

TH1D *m_WW_s5_cheatcluster* = (TH1D)file_s5_icn->Get("m_WW_custom_pairing_nocuts") ;

TH1D *m_WW_s5_icn_udsonly* = (TH1D)file_s5_icn_udsonly->Get("m_WW_uds_nocuts") ;

TH1D *m_ZZ_s5* = (TH1D)file_s5->Get("m_ZZ_nocuts") ;

TH2D *m_m_ZZ_s5* = (TH2D)file_s5_2D->Get("m_m_ZZ_nocuts") ;

TH1D *m_ZZ_s5_icn* = (TH1D)file_s5_icn->Get("m_ZZ_nocuts") ;

TH1D *m_ZZ_s5_cheatcluster* = (TH1D)file_s5_icn->Get("m_ZZ_custom_pairing_nocuts") ;

TH1D *m_ZZ_s5_icn_udsonly* = (TH1D)file_s5_icn_udsonly->Get("m_ZZ_uds_nocuts") ;

TH1D *m_WW_I5_noSLD* = (TH1D)file_I5_SLDs->Get("m_WW_noSLD") ;

TH1D *m_WW_I5_icn_noSLD* = (TH1D)file_I5_SLDs->Get("m_WW_icn_noSLD") ;

TH1D *m_m_WW_I5_icn_noSLD* = (TH1D)file_I5_SLDs_2D->Get("m_m_WW_icn_noSLD") ;

TH1D *m_WW_s5_noSLD* = (TH1D)file_s5_SLDs->Get("m_WW_noSLD") ;

TH1D *m_WW_s5_icn_noSLD* = (TH1D)file_s5_SLDs->Get("m_WW_icn_noSLD") ;

TH1D *m_m_WW_s5_icn_noSLD* = (TH1D)file_s5_SLDs_2D->Get("m_m_WW_icn_noSLD") ;

TH1D *m_ZZ_I5_noSLD* = (TH1D)file_I5_SLDs->Get("m_ZZ_noSLD") ;

TH1D *m_ZZ_I5_icn_noSLD* = (TH1D)file_I5_SLDs->Get("m_ZZ_icn_noSLD") ;

TH1D *m_m_ZZ_I5_icn_noSLD* = (TH1D)file_I5_SLDs_2D->Get("m_m_ZZ_icn_noSLD") ;

TH1D *m_ZZ_s5_noSLD* = (TH1D)file_s5_SLDs->Get("m_ZZ_noSLD") ;

TH1D *m_ZZ_s5_icn_noSLD* = (TH1D)file_s5_SLDs->Get("m_ZZ_icn_noSLD") ;

TH1D *m_m_ZZ_s5_icn_noSLD* = (TH1D)file_s5_SLDs_2D->Get("m_m_ZZ_icn_noSLD") ;

vector deletables {*m_WW_I5*, *m_WW_I5_icn*, *m_WW_I5_icn_udsonly*, *m_ZZ_I5*, *m_ZZ_I5_icn*,
m_ZZ_I5_icn_udsonly, *m_WW_s5*, *m_WW_s5_icn*, *m_WW_s5_icn_udsonly*, *m_ZZ_s5*,

```

m_ZZ_s5_icn, m_ZZ_s5_icn_udsonly, m_WW_I5_noSLD, m_WW_I5_icn_noSLD,
m_WW_s5_noSLD, m_WW_s5_icn_noSLD, m_ZZ_I5_noSLD, m_ZZ_I5_icn_noSLD,
m_ZZ_s5_noSLD, m_ZZ_s5_icn_noSLD, m_m_WW_I5, m_m_ZZ_I5, m_m_WW_s5,
m_m_ZZ_s5, m_m_WW_I5_icn_noSLD, m_m_WW_s5_icn_noSLD, m_m_ZZ_I5_icn_noSLD,
m_m_ZZ_s5_icn_noSLD};

```

```

unique_ptr weak_blue { new TColor( 9024, 5./256., 113./256., 176./256. ) };
unique_ptr dark_blue { new TColor( 9124, 146./256., 197./256., 222./256. ) };
unique_ptr weak_red { new TColor( 9023, 202./256., 0./256., 32./256. ) };
unique_ptr dark_red { new TColor( 9123, 244./256., 165./256., 130./256. ) };

```

```

TImage *ildlogo = TImage::Open("ildlogo.pdf");
deletables.push_back(ildlogo);

```

```

// -----
// TODO Split this sensibly into two plots!
unique_ptr canvas_I5_comp (new TCanvas("canvas_I5_comp", "", 0, 0, 600, 600));
shared_ptr stack_I5_comp = get_comparison_stack( "I5_comp", m_WW_I5_icn_udsonly,
m_ZZ_I5_icn_udsonly, m_WW_I5_icn, m_ZZ_I5_icn, m_WW_I5, m_ZZ_I5 );
stack_I5_comp->Draw("hist nostack");

```

```

unique_ptr leg_I5_comp (new TLegend(0.55, 0.5, 0.9, 0.9));
leg_I5_comp->SetHeader("I5_o1_v02");

```

```

leg_I5_comp->AddEntry(m_WW_I5, "WW", "l");
leg_I5_comp->AddEntry(m_WW_I5_icn, "WW, from icn", "l");
leg_I5_comp->AddEntry(m_WW_I5_icn_udsonly, "WW, from icn, uds only", "l");
leg_I5_comp->AddEntry(m_ZZ_I5, "ZZ", "l");
leg_I5_comp->AddEntry(m_ZZ_I5_icn, "ZZ, from icn", "l");
leg_I5_comp->AddEntry(m_ZZ_I5_icn_udsonly, "ZZ, from icn, uds only", "l");
leg_I5_comp->Draw();

```

```

canvas_I5_comp->SetLeftMargin(0.18);
stack_I5_comp->GetYaxis()->SetTitleOffset(1.34);

```

```

string plot_name_I5_comp = ".I5_comp";
canvas_I5_comp->Print((output_dir + plot_name_I5_comp + ".pdf").c_str());
canvas_I5_comp->Print((output_dir + plot_name_I5_comp + ".C").c_str());
// -----

```

```

// -----
unique_ptr canvas_s5_comp (new TCanvas("canvas_s5_comp", "", 0, 0, 600, 600));
shared_ptr stack_s5_comp = get_comparison_stack( "s5_comp", m_WW_s5_icn_udsonly,

```

```

m_ZZ_s5_icn_udsonly, m_WW_s5_icn, m_ZZ_s5_icn, m_WW_s5, m_ZZ_s5 );
stack_s5_comp->Draw("hist nostack");

unique_ptr leg_s5_comp (new TLegend(0.55, 0.5, 0.9, 0.9));
leg_s5_comp->SetHeader("s5_o1_v02");

leg_s5_comp->AddEntry(m_WW_s5, "WW", "l");
leg_s5_comp->AddEntry(m_WW_s5_icn, "WW, from icn", "l");
leg_s5_comp->AddEntry(m_WW_s5_icn_udsonly, "WW, from icn, uds only", "l");
leg_s5_comp->AddEntry(m_ZZ_s5, "ZZ", "l");
leg_s5_comp->AddEntry(m_ZZ_s5_icn, "ZZ, from icn", "l");
leg_s5_comp->AddEntry(m_ZZ_s5_icn_udsonly, "ZZ, from icn, uds only", "l");
leg_s5_comp->Draw();

canvas_s5_comp->SetLeftMargin(0.18);
stack_s5_comp->GetYaxis()->SetTitleOffset(1.34);

string plot_name_s5_comp = "/s5_comp";
canvas_s5_comp->Print((output_dir + plot_name_s5_comp + ".pdf").c_str());
canvas_s5_comp->Print((output_dir + plot_name_s5_comp + ".C").c_str());
// -----

// -----

shared_ptr canvas_l5_comp_gen (new TCanvas("canvas_l5_comp_gen", "", 0, 0, 1200, 1200));
shared_ptr stack_l5_comp_gen = get_comparison_stack( "l5_comp_gen", m_WW_genlevel,
m_ZZ_genlevel, m_WW_l5_icn, m_ZZ_l5_icn, m_WW_l5, m_ZZ_l5 );
stack_l5_comp_gen->Draw("hist nostack");

unique_ptr leg_l5_comp_gen (new TLegend(0.55, 0.5, 0.9, 0.9));
leg_l5_comp_gen->SetHeader("#splitline{Large ILD,}{after preselection}");

leg_l5_comp_gen->AddEntry(m_WW_genlevel, "WW, MC level", "l");
leg_l5_comp_gen->AddEntry(m_WW_l5_icn, "WW, cheated jets", "l");
leg_l5_comp_gen->AddEntry(m_WW_l5, "WW, full recons.", "l");
leg_l5_comp_gen->AddEntry(m_ZZ_genlevel, "ZZ, MC level", "l");
leg_l5_comp_gen->AddEntry(m_ZZ_l5_icn, "ZZ, cheated jets", "l");
leg_l5_comp_gen->AddEntry(m_ZZ_l5, "ZZ, full recons.", "l");
leg_l5_comp_gen->Draw();

canvas_l5_comp_gen->SetLeftMargin(0.2);
stack_l5_comp_gen->GetYaxis()->SetTitleOffset(1.42);

shared_ptr l5_comp_gen_logo = add_ILD_mark( canvas_l5_comp_gen, 55, 1800, 0.1);

```

```

shared_ptr l5_comp_gen_prelim = mark_preliminary( canvas_l5_comp_gen, 60, 20, 0.2, 45 );

string plot_name_l5_comp_gen = "/l5_comp_gen";
canvas_l5_comp_gen->Print((output_dir + plot_name_l5_comp_gen + ".pdf").c_str());
canvas_l5_comp_gen->Print((output_dir + plot_name_l5_comp_gen + ".jpg").c_str());
canvas_l5_comp_gen->Print((output_dir + plot_name_l5_comp_gen + ".C").c_str());
// -----

// -----

shared_ptr canvas_s5_comp_gen (new TCanvas("canvas_s5_comp_gen", "", 0, 0, 1200, 1200));
shared_ptr stack_s5_comp_gen = get_comparison_stack( "s5_comp_gen", m_WW_genlevel,
m_ZZ_genlevel, m_WW_s5_icn, m_ZZ_s5_icn, m_WW_s5, m_ZZ_s5 );
stack_s5_comp_gen->Draw("hist nostack");

unique_ptr leg_s5_comp_gen (new TLegend(0.55, 0.5, 0.9, 0.9));
leg_s5_comp_gen->SetHeader("#splitline{Small ILD,}{after preselection}");

leg_s5_comp_gen->AddEntry(m_WW_genlevel, "WW, MC level", "l");
leg_s5_comp_gen->AddEntry(m_WW_s5_icn, "WW, cheated jets", "l");
leg_s5_comp_gen->AddEntry(m_WW_s5, "WW, full recons.", "l");
leg_s5_comp_gen->AddEntry(m_ZZ_genlevel, "ZZ, MC level", "l");
leg_s5_comp_gen->AddEntry(m_ZZ_s5_icn, "ZZ, cheated jets", "l");
leg_s5_comp_gen->AddEntry(m_ZZ_s5, "ZZ, full recons.", "l");
leg_s5_comp_gen->Draw();

canvas_s5_comp_gen->SetLeftMargin(0.2);
stack_s5_comp_gen->GetYaxis()->SetTitleOffset(1.42);

shared_ptr s5_comp_gen_logo = add_ILD_mark( canvas_s5_comp_gen, 55, 1800, 0.1);
shared_ptr s5_comp_gen_prelim = mark_preliminary( canvas_s5_comp_gen, 60, 20, 0.2, 45 );

string plot_name_s5_comp_gen = "/s5_comp_gen";
canvas_s5_comp_gen->Print((output_dir + plot_name_s5_comp_gen + ".pdf").c_str());
canvas_s5_comp_gen->Print((output_dir + plot_name_s5_comp_gen + ".jpg").c_str());
canvas_s5_comp_gen->Print((output_dir + plot_name_s5_comp_gen + ".C").c_str());
// -----

// -----

double canvas_l5_comp_cheating_height = 1200;
double canvas_l5_comp_cheating_width = 1300;

shared_ptr canvas_l5_comp_cheating (new TCanvas("canvas_l5_comp_cheating", "", 0, 0,
canvas_l5_comp_cheating_width, canvas_l5_comp_cheating_height));

```



```
shared_ptr stack_I5_comp_cheating = make_shared( "I5_comp_cheating", "; (m_{jj,1} +  
m_{jj,2}))/2 [GeV]; Events" );
```

```
m_WW_I5->SetLineColor(kBlue);  
m_ZZ_I5->SetLineColor(kRed);  
m_WW_I5_cheatcluster->SetLineColor(kBlue-3);  
m_ZZ_I5_cheatcluster->SetLineColor(kRed-3);  
m_WW_I5_icn->SetLineColor(kBlue-6);  
m_ZZ_I5_icn->SetLineColor(kRed-6);
```

```
// m_WW_I5->SetLineStyle(1);  
// m_ZZ_I5->SetLineStyle(1);  
// m_WW_I5_cheatcluster->SetLineStyle(10);  
// m_ZZ_I5_cheatcluster->SetLineStyle(10);  
// m_WW_I5_icn->SetLineStyle(8);  
// m_ZZ_I5_icn->SetLineStyle(8);  
m_WW_I5->SetLineStyle(1);  
m_ZZ_I5->SetLineStyle(1);  
m_WW_I5_cheatcluster->SetLineStyle(2);  
m_ZZ_I5_cheatcluster->SetLineStyle(2);  
m_WW_I5_icn->SetLineStyle(3);  
m_ZZ_I5_icn->SetLineStyle(3);
```

```
m_WW_I5->SetLineWidth(3);  
m_ZZ_I5->SetLineWidth(3);  
m_WW_I5_cheatcluster->SetLineWidth(3);  
m_ZZ_I5_cheatcluster->SetLineWidth(3);  
m_WW_I5_icn->SetLineWidth(3);  
m_ZZ_I5_icn->SetLineWidth(3);
```

```
stack_I5_comp_cheating->Add(m_WW_I5_icn);  
stack_I5_comp_cheating->Add(m_ZZ_I5_icn);  
stack_I5_comp_cheating->Add(m_WW_I5_cheatcluster);  
stack_I5_comp_cheating->Add(m_ZZ_I5_cheatcluster);  
stack_I5_comp_cheating->Add(m_WW_I5);  
stack_I5_comp_cheating->Add(m_ZZ_I5);  
stack_I5_comp_cheating->Draw("axis"); // Draw only axis
```

```
unique_ptr leg_I5_comp_cheating (new TLegend(0.6, 0.42, 0.9, 0.8));  
leg_I5_comp_cheating->SetHeader("IDR-L");  
TLine *full_line_I5_comp_cheating = new TLine();  
TLine *dots_line_I5_comp_cheating = new TLine();  
TLine *dash_line_I5_comp_cheating = new TLine();
```

```

full_line_I5_comp_cheating->SetLineStyle(1); full_line_I5_comp_cheating->SetLineWidth(3);
full_line_I5_comp_cheating->SetLineColor(1); full_line_I5_comp_cheating->Draw();
dots_line_I5_comp_cheating->SetLineStyle(2); dots_line_I5_comp_cheating->SetLineWidth(3);
dots_line_I5_comp_cheating->SetLineColor(1); dots_line_I5_comp_cheating->Draw();
dash_line_I5_comp_cheating->SetLineStyle(3); dash_line_I5_comp_cheating->SetLineWidth(3);
dash_line_I5_comp_cheating->SetLineColor(1); dash_line_I5_comp_cheating->Draw();
deletables.push_back(full_line_I5_comp_cheating);
deletables.push_back(dots_line_I5_comp_cheating);
deletables.push_back(dash_line_I5_comp_cheating);
leg_I5_comp_cheating->AddEntry(full_line_I5_comp_cheating, "#splitline{full}{reconstruction}",
"I");
leg_I5_comp_cheating->AddEntry(dots_line_I5_comp_cheating, "#splitline{ideal}{clustering}", "I");
leg_I5_comp_cheating->AddEntry(dash_line_I5_comp_cheating, "#splitline{ideal clust.}{&
pairing}", "I");
leg_I5_comp_cheating->Draw();

```

```

stack_I5_comp_cheating->Draw("hist nostack same");
double I5_comp_cheating_old_left_margin = canvas_I5_comp_cheating->GetLeftMargin();
adjust_canvas_horizontal_to_square_pad(canvas_I5_comp_cheating);
stack_I5_comp_cheating->GetYaxis()->SetTitleOffset( stack_I5_comp_cheating->GetYaxis()-
>GetTitleOffset() * canvas_I5_comp_cheating-
>GetLeftMargin()/I5_comp_cheating_old_left_margin );
stack_I5_comp_cheating->SetMaximum( 1.15 * m_WW_I5_icn->GetMaximum() );

```

```

shared_ptr I5_comp_cheating_logo = add_ILD_mark( canvas_I5_comp_cheating, 55, 1.03 *
m_WW_I5_icn->GetMaximum(), 0.1);
shared_ptr I5_comp_cheating_prelim = add_prelim_mark( canvas_I5_comp_cheating, 71, 1.03 *
m_WW_I5_icn->GetMaximum(), 0.07);

```

```

string plot_name_I5_comp_cheating = ".I5_comp_cheating";
canvas_I5_comp_cheating->Print((output_dir + plot_name_I5_comp_cheating + ".pdf").c_str());
canvas_I5_comp_cheating->Print((output_dir + plot_name_I5_comp_cheating + ".jpg").c_str());
canvas_I5_comp_cheating->Print((output_dir + plot_name_I5_comp_cheating + ".C").c_str());
// -----

```

```

// -----
double canvas_I5_m_height = 1200;
double canvas_I5_m_width = 1250;

```

```

shared_ptr canvas_I5_m (new TCanvas("canvas_I5_m", "", 0, 0, canvas_I5_m_width,
canvas_I5_m_height));
shared_ptr stack_I5_m = make_shared( "I5_m", "; (m_{jj,1} + m_{jj,2})/2 [GeV]; Events" );

```

```

m_WW_I5->SetLineColor(kBlue);
m_ZZ_I5->SetLineColor(kRed);
m_WW_I5->SetLineStyle(1);
m_ZZ_I5->SetLineStyle(1);
m_WW_I5->SetLineWidth(3);
m_ZZ_I5->SetLineWidth(3);

stack_I5_m->Add(m_WW_I5);
stack_I5_m->Add(m_ZZ_I5);
stack_I5_m->Draw("axis"); // Draw only axis

unique_ptr leg_I5_m (new TLegend(0.6, 0.55, 0.9, 0.8));
leg_I5_m->SetHeader("#splitline{IDR-L}{full reconstr.}");
leg_I5_m->AddEntry(m_WW_I5, "#font[12]{WW} signal", "l");
leg_I5_m->AddEntry(m_ZZ_I5, "#font[12]{ZZ} signal", "l");
leg_I5_m->Draw();

stack_I5_m->Draw("hist nostack same");
double I5_m_old_left_margin = canvas_I5_m->GetLeftMargin();
adjust_canvas_horizontal_to_square_pad(canvas_I5_m);
stack_I5_m->GetYaxis()->SetTitleOffset( stack_I5_m->GetYaxis()->GetTitleOffset() *
canvas_I5_m->GetLeftMargin()/I5_m_old_left_margin );
stack_I5_m->SetMaximum( 1.15 * m_WW_I5->GetMaximum() );

shared_ptr I5_m_logo = add_ILD_mark( canvas_I5_m, 55, 1.03 * m_WW_I5->GetMaximum(),
0.1);
shared_ptr I5_m_prelim = add_prelim_mark( canvas_I5_m, 71, 1.03 * m_WW_I5-
>GetMaximum(), 0.07);

string plot_name_I5_m = ".I5_m";
canvas_I5_m->Print((output_dir + plot_name_I5_m + ".pdf").c_str());
canvas_I5_m->Print((output_dir + plot_name_I5_m + ".jpg").c_str());
canvas_I5_m->Print((output_dir + plot_name_I5_m + ".C").c_str());
// -----

// Should be same for all of these plots
int canvas_height = 1600;
int canvas_width = 1200;

double pad_border = (canvas_width0.74 + canvas_height0.08) / canvas_height; // Chosen to
make upper plot square (not sure if it entirely worked)
double canvas_size_ratio = (pad_border)/(1.0-pad_border);

```

```

Double_t logo_x0 = 0.25;
Double_t logo_y0 = 0.85;

Double_t prelim_x0 = 0.3;
Double_t prelim_y0 = 0.1;

// -----
shared_ptr canvas_ls_comp_rec (new TCanvas("canvas_ls_comp_rec", "", 0, 0, canvas_width,
canvas_height));
canvas_ls_comp_rec->SetLeftMargin(0.18);

unique_ptr pad_ls_comp_rec_1 (new TPad("pad_ls_comp_rec_1", "pad_ls_comp_rec_1", 0, 1.0-
pad_border, 1, 1.0));
pad_ls_comp_rec_1->SetBottomMargin(0); // Upper and lower plot are joined
pad_ls_comp_rec_1->Draw(); // Draw the upper pad: pad_ls_comp_rec_1
canvas_ls_comp_rec->cd();
unique_ptr pad_ls_comp_rec_2 (new TPad("pad_ls_comp_rec_2", "pad_ls_comp_rec_2", 0, 0,
1, 1.0-pad_border));
pad_ls_comp_rec_2->SetTopMargin(0);
pad_ls_comp_rec_2->SetBottomMargin(0.3);
pad_ls_comp_rec_2->Draw();

pad_ls_comp_rec_1->cd(); // pad_ls_comp_rec_1 becomes the current pad
shared_ptr stack_ls_comp_rec = get_comparison_stack( "ls_comp_rec", m_WW_I5, m_ZZ_I5,
m_WW_s5, m_ZZ_s5 );
stack_ls_comp_rec->SetMaximum(820);
stack_ls_comp_rec->Draw("hist nostack");
// stack_ls_comp_rec->GetYaxis()->SetTitleOffset(1.34);

unique_ptr leg_ls_comp_rec (new TLegend(0.6, 0.6, 0.9, 0.9));
leg_ls_comp_rec->SetHeader("Full reconstruction");

leg_ls_comp_rec->AddEntry(m_WW_I5, "WW, large ILD", "l");
leg_ls_comp_rec->AddEntry(m_ZZ_I5, "ZZ, large ILD", "l");
leg_ls_comp_rec->AddEntry(m_WW_s5, "WW, small ILD", "l");
leg_ls_comp_rec->AddEntry(m_ZZ_s5, "ZZ, small ILD", "l");
leg_ls_comp_rec->Draw();

pad_ls_comp_rec_2->cd(); // pad_ls_comp_rec_2 becomes the current pad

shared_ptr stack_ls_comp_rec_ratio = get_ratio_stack( "stack_ls_comp_rec_ratio", m_WW_I5,
m_ZZ_I5, m_WW_s5, m_ZZ_s5, "large/small" );
stack_ls_comp_rec_ratio->Draw("e2 p nostack");

```

```

stack_ls_comp_rec_ratio->SetMinimum(0);
stack_ls_comp_rec_ratio->SetMaximum(2);

renormalize_axis_sizes( stack_ls_comp_rec_ratio, canvas_size_ratio );

unique_ptr line_ls_comp_rec_ratio { new TLine( 50, 1, 120, 1) };
line_ls_comp_rec_ratio->Draw();

shared_ptr ls_comp_rec_logo = add_ILD_mark( canvas_ls_comp_rec, logo_x0, logo_y0, 0.1);
shared_ptr ls_comp_rec_prelim = mark_preliminary( canvas_ls_comp_rec, prelim_x0, prelim_y0
);

string plot_name_ls_comp_rec = "./ls_comp_rec";
canvas_ls_comp_rec->Print((output_dir + plot_name_ls_comp_rec + ".pdf").c_str());
canvas_ls_comp_rec->Print((output_dir + plot_name_ls_comp_rec + ".jpg").c_str());
canvas_ls_comp_rec->Print((output_dir + plot_name_ls_comp_rec + ".C").c_str());
// -----

// -----
shared_ptr canvas_ls_comp_icn (new TCanvas("canvas_ls_comp_icn", "", 0, 0, canvas_width,
canvas_height));
canvas_ls_comp_icn->SetLeftMargin(0.18);

unique_ptr pad_ls_comp_icn_1 (new TPad("pad_ls_comp_icn_1", "pad_ls_comp_icn_1", 0, 1.0-
pad_border, 1, 1.0));
pad_ls_comp_icn_1->SetBottomMargin(0); // Upper and lower plot are joined
pad_ls_comp_icn_1->Draw();
canvas_ls_comp_icn->cd();
unique_ptr pad_ls_comp_icn_2 (new TPad("pad_ls_comp_icn_2", "pad_ls_comp_icn_2", 0, 0, 1,
1.0-pad_border));
pad_ls_comp_icn_2->SetTopMargin(0);
pad_ls_comp_icn_2->SetBottomMargin(0.3);
pad_ls_comp_icn_2->Draw();

pad_ls_comp_icn_1->cd();
shared_ptr stack_ls_comp_icn = get_comparison_stack( "ls_comp_icn", m_WW_I5_icn,
m_ZZ_I5_icn, m_WW_s5_icn, m_ZZ_s5_icn );
stack_ls_comp_icn->SetMaximum(820);
stack_ls_comp_icn->Draw("hist nostack");

unique_ptr leg_ls_comp_icn (new TLegend(0.6, 0.6, 0.9, 0.9));
leg_ls_comp_icn->SetHeader("#splitline{Jet clustering/}{pairing cheated}");

```

```

leg_ls_comp_icn->AddEntry(m_WW_I5, "WW, large ILD", "l");
leg_ls_comp_icn->AddEntry(m_ZZ_I5, "ZZ, large ILD", "l");
leg_ls_comp_icn->AddEntry(m_WW_s5, "WW, small ILD", "l");
leg_ls_comp_icn->AddEntry(m_ZZ_s5, "ZZ, small ILD", "l");
leg_ls_comp_icn->Draw();

pad_ls_comp_icn_2->cd();

shared_ptr stack_ls_comp_icn_ratio = get_ratio_stack( "stack_ls_comp_icn_ratio",
m_WW_I5_icn, m_ZZ_I5_icn, m_WW_s5_icn, m_ZZ_s5_icn, "large/small" );
stack_ls_comp_icn_ratio->Draw("e2 p nostack");
stack_ls_comp_icn_ratio->SetMinimum(0);
stack_ls_comp_icn_ratio->SetMaximum(2);
renormalize_axis_sizes( stack_ls_comp_icn_ratio, canvas_size_ratio );

unique_ptr line_ls_comp_icn_ratio { new TLine( 50, 1, 120, 1) };
line_ls_comp_icn_ratio->Draw();

shared_ptr ls_comp_icn_logo = add_ILD_mark( canvas_ls_comp_icn, logo_x0, logo_y0, 0.1);
shared_ptr ls_comp_icn_prelim = mark_preliminary( canvas_ls_comp_icn, prelim_x0, prelim_y0
);

string plot_name_ls_comp_icn = ".ls_comp_icn";
canvas_ls_comp_icn->Print((output_dir + plot_name_ls_comp_icn + ".pdf").c_str());
canvas_ls_comp_icn->Print((output_dir + plot_name_ls_comp_icn + ".jpg").c_str());
canvas_ls_comp_icn->Print((output_dir + plot_name_ls_comp_icn + ".C").c_str());
// -----

// -----

shared_ptr canvas_ls_comp_icn_uds (new TCanvas("canvas_ls_comp_icn_uds", "", 0, 0,
canvas_width, canvas_height));
canvas_ls_comp_icn_uds->SetLeftMargin(0.18);

unique_ptr pad_ls_comp_icn_uds_1 (new TPad("pad_ls_comp_icn_uds_1",
"pad_ls_comp_icn_uds_1", 0, 1.0-pad_border, 1, 1.0));
pad_ls_comp_icn_uds_1->SetBottomMargin(0); // Upper and lower plot are joined
pad_ls_comp_icn_uds_1->Draw();
canvas_ls_comp_icn_uds->cd();
unique_ptr pad_ls_comp_icn_uds_2 (new TPad("pad_ls_comp_icn_uds_2",
"pad_ls_comp_icn_uds_2", 0, 0, 1, 1.0-pad_border));
pad_ls_comp_icn_uds_2->SetTopMargin(0);
pad_ls_comp_icn_uds_2->SetBottomMargin(0.3);
pad_ls_comp_icn_uds_2->Draw();

```

```

pad_ls_comp_icn_uds_1->cd();
shared_ptr stack_ls_comp_icn_uds = get_comparison_stack( "ls_comp_icn_uds",
m_WW_I5_icn_udsonly, m_ZZ_I5_icn_udsonly, m_WW_s5_icn_udsonly, m_ZZ_s5_icn_udsonly
);
stack_ls_comp_icn_uds->SetMaximum(820);
stack_ls_comp_icn_uds->Draw("hist nostack");

unique_ptr leg_ls_comp_icn_uds (new TLegend(0.62, 0.6, 0.92, 0.9));
leg_ls_comp_icn_uds->SetHeader("#splitline{Jet cl./pair. cheated,}{only light q events}");

leg_ls_comp_icn_uds->AddEntry(m_WW_I5, "WW, large ILD", "l");
leg_ls_comp_icn_uds->AddEntry(m_ZZ_I5, "ZZ, large ILD", "l");
leg_ls_comp_icn_uds->AddEntry(m_WW_s5, "WW, small ILD", "l");
leg_ls_comp_icn_uds->AddEntry(m_ZZ_s5, "ZZ, small ILD", "l");
leg_ls_comp_icn_uds->Draw();

pad_ls_comp_icn_uds_2->cd();

shared_ptr stack_ls_comp_icn_uds_ratio = get_ratio_stack( "stack_ls_comp_icn_uds_ratio",
m_WW_I5_icn_udsonly, m_ZZ_I5_icn_udsonly, m_WW_s5_icn_udsonly, m_ZZ_s5_icn_udsonly,
"large/small" );
stack_ls_comp_icn_uds_ratio->Draw("e2 p nostack");
stack_ls_comp_icn_uds_ratio->SetMinimum(0);
stack_ls_comp_icn_uds_ratio->SetMaximum(2);
renormalize_axis_sizes( stack_ls_comp_icn_uds_ratio, canvas_size_ratio );

unique_ptr line_ls_comp_icn_uds_ratio { new TLine( 50, 1, 120, 1) };
line_ls_comp_icn_uds_ratio->Draw();

shared_ptr ls_comp_icn_uds_logo = add_ILD_mark( canvas_ls_comp_icn_uds, logo_x0,
logo_y0, 0.1);
shared_ptr ls_comp_icn_uds_prelim = mark_preliminary( canvas_ls_comp_icn_uds, prelim_x0,
prelim_y0 );

string plot_name_ls_comp_icn_uds = "./ls_comp_icn_uds";
canvas_ls_comp_icn_uds->Print((output_dir + plot_name_ls_comp_icn_uds + ".pdf").c_str());
canvas_ls_comp_icn_uds->Print((output_dir + plot_name_ls_comp_icn_uds + ".jpg").c_str());
canvas_ls_comp_icn_uds->Print((output_dir + plot_name_ls_comp_icn_uds + ".C").c_str());

// -----

shared_ptr canvas_comp_final (new TCanvas("canvas_comp_final", "", 0, 0, 1200, 1200));
shared_ptr stack_comp_final = make_shared( "comp_final", "; (m_{jj,1} + m_{jj,2})/2 [GeV];

```

```

Events" );
stack_comp_final->Add(m_WW_I5_icn);
stack_comp_final->Add(m_WW_I5);
stack_comp_final->Add(m_WW_s5_icn);
stack_comp_final->Add(m_WW_s5);
stack_comp_final->Add(m_ZZ_I5_icn);
stack_comp_final->Add(m_ZZ_I5);
stack_comp_final->Add(m_ZZ_s5_icn);
stack_comp_final->Add(m_ZZ_s5);

m_WW_I5_icn->SetLineStyle(1);
m_WW_I5->SetLineStyle(1);
m_WW_s5_icn->SetLineStyle(2);
m_WW_s5->SetLineStyle(2);
m_ZZ_I5_icn->SetLineStyle(1);
m_ZZ_I5->SetLineStyle(1);
m_ZZ_s5_icn->SetLineStyle(2);
m_ZZ_s5->SetLineStyle(2);

m_WW_I5_icn->SetLineColorAlpha(kBlue-6, 0.85);
m_WW_s5_icn->SetLineColorAlpha(kBlue-8, 0.85);
m_ZZ_I5_icn->SetLineColorAlpha(kRed-8, 0.85);
m_ZZ_s5_icn->SetLineColorAlpha(kRed-10, 0.85);
m_WW_I5->SetLineColorAlpha(kBlue, 1.0);
m_WW_s5->SetLineColorAlpha(kBlue+1, 0.95);
m_ZZ_I5->SetLineColorAlpha(kRed, 1.0);
m_ZZ_s5->SetLineColorAlpha(kRed+1, 0.95);

stack_comp_final->SetMaximum(1850);
stack_comp_final->Draw("hist nostack");

unique_ptr leg_comp_final (new TLegend(0.6, 0.6, 0.9, 0.8));
TLine *full_line = new TLine();
TLine *dash_line = new TLine();
full_line->SetLineStyle(1); full_line->SetLineWidth(5); full_line->Draw();
dash_line->SetLineStyle(2); dash_line->SetLineWidth(5); dash_line->SetLineColor(kGray+3);
dash_line->Draw();
leg_comp_final->SetHeader("full reco");
leg_comp_final->AddEntry(full_line, "IDR-L", "l");
leg_comp_final->AddEntry(dash_line, "IDR-S", "l");
leg_comp_final->Draw();
deletables.push_back(full_line);
deletables.push_back(dash_line);

```



```

shared_ptr cheated_level_tex (new TLatex(90,700,"#splitline{light: cheated}{#color[0]{light:
}jets}"));
cheated_level_tex->SetName( (string(canvas_comp_final->GetName()) +
"_cheated_level_tex").c_str() );
cheated_level_tex->SetTextSize(0.045);
cheated_level_tex->SetFont(62);
cheated_level_tex->SetLineWidth(2);
cheated_level_tex->Draw();

shared_ptr reco_level_tex (new TLatex(90,880,"#splitline{dark: full reco}{})");
reco_level_tex->SetName( (string(canvas_comp_final->GetName()) + "_reco_level_tex").c_str()
);
reco_level_tex->SetTextSize(0.045);
reco_level_tex->SetFont(62);
reco_level_tex->SetLineWidth(2);
reco_level_tex->Draw();

canvas_comp_final->SetLeftMargin(0.2);
stack_comp_final->GetYaxis()->SetTitleOffset(1.42);

shared_ptr comp_final_logo = add_ILD_mark( canvas_comp_final, 55, 1600, 0.1);
shared_ptr comp_final_prelim = add_prelim_mark( canvas_comp_final, 71, 1600, 0.07);
//
string plot_name_comp_final = "./comp_final";
canvas_comp_final->Print((output_dir + plot_name_comp_final + ".pdf").c_str());
canvas_comp_final->Print((output_dir + plot_name_comp_final + ".jpg").c_str());
canvas_comp_final->Print((output_dir + plot_name_comp_final + ".C").c_str());

// -----

shared_ptr canvas_comp_final_noSLD (new TCanvas("canvas_comp_final_noSLD", "", 0, 0,
1200, 1200));
shared_ptr stack_comp_final_noSLD = make_shared( "comp_final_noSLD", "; (m_{jj,1} +
m_{jj,2})/2 [GeV]; Events" );
stack_comp_final_noSLD->Add(m_WW_l5_icn_noSLD);
stack_comp_final_noSLD->Add(m_WW_l5_noSLD);
stack_comp_final_noSLD->Add(m_WW_s5_icn_noSLD);
stack_comp_final_noSLD->Add(m_WW_s5_noSLD);
stack_comp_final_noSLD->Add(m_ZZ_l5_icn_noSLD);
stack_comp_final_noSLD->Add(m_ZZ_l5_noSLD);
stack_comp_final_noSLD->Add(m_ZZ_s5_icn_noSLD);
stack_comp_final_noSLD->Add(m_ZZ_s5_noSLD);

```

```

m_WW_I5_icn_noSLD->SetLineStyle(1);
m_WW_I5_noSLD->SetLineStyle(1);
m_WW_s5_icn_noSLD->SetLineStyle(2);
m_WW_s5_noSLD->SetLineStyle(2);
m_ZZ_I5_icn_noSLD->SetLineStyle(1);
m_ZZ_I5_noSLD->SetLineStyle(1);
m_ZZ_s5_icn_noSLD->SetLineStyle(2);
m_ZZ_s5_noSLD->SetLineStyle(2);

```

```

m_WW_I5_icn_noSLD->SetLineColorAlpha(kBlue-6, 0.85);
m_WW_s5_icn_noSLD->SetLineColorAlpha(kBlue-8, 0.85);
m_ZZ_I5_icn_noSLD->SetLineColorAlpha(kRed-8, 0.85);
m_ZZ_s5_icn_noSLD->SetLineColorAlpha(kRed-10, 0.85);
m_WW_I5_noSLD->SetLineColorAlpha(kBlue, 1.0);
m_WW_s5_noSLD->SetLineColorAlpha(kBlue+1, 0.95);
m_ZZ_I5_noSLD->SetLineColorAlpha(kRed, 1.0);
m_ZZ_s5_noSLD->SetLineColorAlpha(kRed+1, 0.95);

```

```

stack_comp_final_noSLD->SetMaximum(1850);
stack_comp_final_noSLD->Draw("hist nostack");

```

```

unique_ptr leg_comp_final_noSLD (new TLegend(0.6, 0.55, 0.9, 0.8));
TLine *full_line_noSLD = new TLine();
TLine *dash_line_noSLD = new TLine();
full_line_noSLD->SetLineStyle(1); full_line_noSLD->SetLineWidth(5); full_line_noSLD->Draw();
dash_line_noSLD->SetLineStyle(2); dash_line_noSLD->SetLineWidth(5); dash_line_noSLD->SetLineColor(kGray+3); dash_line_noSLD->Draw();
leg_comp_final_noSLD->SetHeader("#splitline{no semi-lep.}{decays}");
leg_comp_final_noSLD->AddEntry(full_line_noSLD, "IDR-L", "l");
leg_comp_final_noSLD->AddEntry(dash_line_noSLD, "IDR-S", "l");
leg_comp_final_noSLD->Draw();
deletables.push_back(full_line_noSLD);
deletables.push_back(dash_line_noSLD);

```

```

shared_ptr cheated_level_tex_noSLD (new TLatex(90,560,"#splitline{light: cheated}{{color[0]
{light: }jets}}"));
cheated_level_tex_noSLD->SetName( (string(canvas_comp_final_noSLD->GetName()) +
"_cheated_level_tex_noSLD").c_str() );
cheated_level_tex_noSLD->SetTextSize(0.045);
cheated_level_tex_noSLD->SetTextFont(62);
cheated_level_tex_noSLD->SetLineWidth(2);
cheated_level_tex_noSLD->Draw();

```

```

shared_ptr reco_level_tex_noSLD (new TLatex(90,740,"#splitline{dark: full reco}{}"));
reco_level_tex_noSLD->SetName( (string(canvas_comp_final_noSLD->GetName()) +
"_reco_level_tex_noSLD").c_str() );
reco_level_tex_noSLD->SetTextSize(0.045);
reco_level_tex_noSLD->SetTextFont(62);
reco_level_tex_noSLD->SetLineWidth(2);
reco_level_tex_noSLD->Draw();

canvas_comp_final_noSLD->SetLeftMargin(0.2);
stack_comp_final_noSLD->GetYaxis()->SetTitleOffset(1.42);

shared_ptr comp_final_noSLD_logo = add_ILD_mark( canvas_comp_final_noSLD, 55, 1600,
0.1);
shared_ptr comp_final_noSLD_prelim = add_prelim_mark( canvas_comp_final_noSLD, 71,
1600, 0.07);
//
string plot_name_comp_final_noSLD = "./comp_final_noSLD";
canvas_comp_final_noSLD->Print((output_dir + plot_name_comp_final_noSLD + ".pdf").c_str());
canvas_comp_final_noSLD->Print((output_dir + plot_name_comp_final_noSLD + ".jpg").c_str());
canvas_comp_final_noSLD->Print((output_dir + plot_name_comp_final_noSLD + ".C").c_str());

// -----
// -----
// -----
// IDR FINALISTS
// -----
// -----
// -----

// -----
double canvas_ls_comp_rec_monly_height = 1200;
double canvas_ls_comp_rec_monly_width = 1250;

shared_ptr canvas_ls_comp_rec_monly (new TCanvas("canvas_ls_comp_rec_monly", "", 0, 0,
canvas_ls_comp_rec_monly_width, canvas_ls_comp_rec_monly_height));
shared_ptr stack_ls_comp_rec_monly = make_shared( "ls_comp_rec_monly", "; (m_{jj,1} +
m_{jj,2})/2 [GeV]; Events" );

m_WW_I5->SetLineColor(kBlue); m_WW_I5->SetLineStyle(1); m_WW_I5->SetLineWidth(3);
m_ZZ_I5->SetLineColor(kRed); m_ZZ_I5->SetLineStyle(1); m_ZZ_I5->SetLineWidth(3);
m_WW_s5->SetLineColor(9024); m_WW_s5->SetLineStyle(7); m_WW_s5->SetLineWidth(3);
m_ZZ_s5->SetLineColor(9023); m_ZZ_s5->SetLineStyle(7); m_ZZ_s5->SetLineWidth(3);

```

```
stack_ls_comp_rec_monly->Add(m_WW_I5); stack_ls_comp_rec_monly->Add(m_ZZ_I5);
stack_ls_comp_rec_monly->Add(m_WW_s5); stack_ls_comp_rec_monly->Add(m_ZZ_s5);
```

```
stack_ls_comp_rec_monly->Draw("axis"); // Draw only axis
```

```
unique_ptr leg_ls_comp_rec_monly (new TLegend(0.6, 0.5, 0.97, 0.8));
TLine *full_line_ls_comp_rec_monly = new TLine();
TLine *dash_line_ls_comp_rec_monly = new TLine();
full_line_ls_comp_rec_monly->SetLineStyle(1); full_line_ls_comp_rec_monly->SetLineWidth(3);
full_line_ls_comp_rec_monly->SetLineColor(1); full_line_ls_comp_rec_monly->Draw();
dash_line_ls_comp_rec_monly->SetLineStyle(7); dash_line_ls_comp_rec_monly->
>SetLineWidth(3); dash_line_ls_comp_rec_monly->SetLineColor(1);
dash_line_ls_comp_rec_monly->Draw();
deletables.push_back(full_line_ls_comp_rec_monly);
deletables.push_back(dash_line_ls_comp_rec_monly);
leg_ls_comp_rec_monly->SetHeader("full reconstr.");
leg_ls_comp_rec_monly->AddEntry(full_line_ls_comp_rec_monly, "IDR-L", "l");
leg_ls_comp_rec_monly->AddEntry(dash_line_ls_comp_rec_monly, "IDR-S", "l");
leg_ls_comp_rec_monly->Draw();
```

```
stack_ls_comp_rec_monly->Draw("axis same"); // Draw only axis
```

```
stack_ls_comp_rec_monly->Draw("hist nostack same");
double ls_comp_rec_monly_old_left_margin = canvas_ls_comp_rec_monly->GetLeftMargin();
adjust_canvas_horizontal_to_square_pad(canvas_ls_comp_rec_monly);
stack_ls_comp_rec_monly->GetYaxis()->SetTitleOffset( stack_ls_comp_rec_monly->GetYaxis()-
>GetTitleOffset() * canvas_ls_comp_rec_monly-
>GetLeftMargin()/ls_comp_rec_monly_old_left_margin );
stack_ls_comp_rec_monly->SetMaximum( 1.15 * m_WW_I5->GetMaximum() );
```

```
shared_ptr ls_comp_rec_monly_logo = add_ILD_mark( canvas_ls_comp_rec_monly, 55, 1.03 *
m_WW_I5->GetMaximum(), 0.1);
shared_ptr ls_comp_rec_monly_prelim = add_prelim_mark( canvas_ls_comp_rec_monly, 71,
1.03 * m_WW_I5->GetMaximum(), 0.07);
```

```
string plot_name_ls_comp_rec_monly = "./ls_comp_rec_monly";
canvas_ls_comp_rec_monly->Print((output_dir + plot_name_ls_comp_rec_monly +
".pdf").c_str());
canvas_ls_comp_rec_monly->Print((output_dir + plot_name_ls_comp_rec_monly +
".jpg").c_str());
canvas_ls_comp_rec_monly->Print((output_dir + plot_name_ls_comp_rec_monly + ".C").c_str());
// -----
```

```

// -----
double canvas_m_m_rec_height = 1200;
double canvas_m_m_rec_width = 1250;

shared_ptr canvas_m_m_rec (new TCanvas("canvas_m_m_rec", "", 0, 0,
canvas_m_m_rec_width, canvas_m_m_rec_height));

m_m_WW_I5->SetLineColor(kBlue); m_m_WW_I5->SetLineStyle(1); m_m_WW_I5-
>SetLineWidth(3); m_m_WW_I5->Scale(1.0/m_m_WW_I5->Integral());
m_m_ZZ_I5->SetLineColor(kRed); m_m_ZZ_I5->SetLineStyle(1); m_m_ZZ_I5-
>SetLineWidth(3); m_m_ZZ_I5->Scale(1.0/m_m_ZZ_I5->Integral());
m_m_ZZ_I5->SetTitle("");
m_m_ZZ_I5->Draw("box");
m_m_WW_I5->Draw("box same");

unique_ptr leg_m_m_rec (new TLegend(0.6, 0.5, 0.97, 0.8));
leg_m_m_rec->SetHeader("full reconstr., IDR-L");
leg_m_m_rec->Draw();

// double m_m_rec_old_left_margin = canvas_m_m_rec->GetLeftMargin();
// adjust_canvas_horizontal_to_square_pad(canvas_m_m_rec);
// stack_m_m_rec->GetYaxis()->SetTitleOffset( stack_m_m_rec->GetYaxis()->GetTitleOffset() *
canvas_m_m_rec->GetLeftMargin()/m_m_rec_old_left_margin );
// stack_m_m_rec->SetMaximum( 1.15 * m_m_WW_I5->GetMaximum() );

shared_ptr m_m_rec_logo = add_ILD_mark( canvas_m_m_rec, 40, 120, 0.1);
shared_ptr m_m_rec_prelim = add_prelim_mark( canvas_m_m_rec, 0, 0, 0.07);

string plot_name_m_m_rec = "./m_m_rec";
canvas_m_m_rec->Print((output_dir + plot_name_m_m_rec + ".pdf").c_str());
canvas_m_m_rec->Print((output_dir + plot_name_m_m_rec + ".jpg").c_str());
canvas_m_m_rec->Print((output_dir + plot_name_m_m_rec + ".C").c_str());
// -----

// -----
double canvas_ls_comp_icn_noSLD_monly_height = 1200;
double canvas_ls_comp_icn_noSLD_monly_width = 1250;

shared_ptr canvas_ls_comp_icn_noSLD_monly (new
TCanvas("canvas_ls_comp_icn_noSLD_monly", "", 0, 0,
canvas_ls_comp_icn_noSLD_monly_width, canvas_ls_comp_icn_noSLD_monly_height));
shared_ptr stack_ls_comp_icn_noSLD_monly = make_shared( "ls_comp_icn_noSLD_monly", ";
(m_{jj,1} + m_{jj,2})/2 [GeV]; Events" );

```

```

m_WW_I5_icn_noSLD->SetLineColor(kBlue); m_WW_I5_icn_noSLD->SetLineStyle(1);
m_WW_I5_icn_noSLD->SetLineWidth(3);
m_ZZ_I5_icn_noSLD->SetLineColor(kRed); m_ZZ_I5_icn_noSLD->SetLineStyle(1);
m_ZZ_I5_icn_noSLD->SetLineWidth(3);
m_WW_s5_icn_noSLD->SetLineColor(9024); m_WW_s5_icn_noSLD->SetLineStyle(7);
m_WW_s5_icn_noSLD->SetLineWidth(3);
m_ZZ_s5_icn_noSLD->SetLineColor(9023); m_ZZ_s5_icn_noSLD->SetLineStyle(7);
m_ZZ_s5_icn_noSLD->SetLineWidth(3);

stack_Is_comp_icn_noSLD_monly->Add(m_WW_I5_icn_noSLD);
stack_Is_comp_icn_noSLD_monly->Add(m_ZZ_I5_icn_noSLD);
stack_Is_comp_icn_noSLD_monly->Add(m_WW_s5_icn_noSLD);
stack_Is_comp_icn_noSLD_monly->Add(m_ZZ_s5_icn_noSLD);

stack_Is_comp_icn_noSLD_monly->Draw("axis"); // Draw only axis

unique_ptr leg_Is_comp_icn_noSLD_monly (new TLegend(0.6, 0.5, 0.97, 0.8));
TLine *full_line_Is_comp_icn_noSLD_monly = new TLine();
TLine *dash_line_Is_comp_icn_noSLD_monly = new TLine();
full_line_Is_comp_icn_noSLD_monly->SetLineStyle(1); full_line_Is_comp_icn_noSLD_monly-
>SetLineWidth(3); full_line_Is_comp_icn_noSLD_monly->SetLineColor(1);
full_line_Is_comp_icn_noSLD_monly->Draw();
dash_line_Is_comp_icn_noSLD_monly->SetLineStyle(7);
dash_line_Is_comp_icn_noSLD_monly->SetLineWidth(3);
dash_line_Is_comp_icn_noSLD_monly->SetLineColor(1);
dash_line_Is_comp_icn_noSLD_monly->Draw();
deletables.push_back(full_line_Is_comp_icn_noSLD_monly);
deletables.push_back(dash_line_Is_comp_icn_noSLD_monly);
leg_Is_comp_icn_noSLD_monly->SetHeader("full icn_noSLDonstr.");
leg_Is_comp_icn_noSLD_monly->AddEntry(full_line_Is_comp_icn_noSLD_monly, "IDR-L", "l");
leg_Is_comp_icn_noSLD_monly->AddEntry(dash_line_Is_comp_icn_noSLD_monly, "IDR-S", "l");
leg_Is_comp_icn_noSLD_monly->Draw();

stack_Is_comp_icn_noSLD_monly->Draw("axis same"); // Draw only axis

stack_Is_comp_icn_noSLD_monly->Draw("hist nostack same");
double Is_comp_icn_noSLD_monly_old_left_margin = canvas_Is_comp_icn_noSLD_monly-
>GetLeftMargin();
adjust_canvas_horizontal_to_square_pad(canvas_Is_comp_icn_noSLD_monly);
stack_Is_comp_icn_noSLD_monly->GetYaxis()->SetTitleOffset(
stack_Is_comp_icn_noSLD_monly->GetYaxis()->GetTitleOffset() *
canvas_Is_comp_icn_noSLD_monly-
>GetLeftMargin()/Is_comp_icn_noSLD_monly_old_left_margin );

```

```

stack_ls_comp_icn_noSLD_monly->SetMaximum( 1.15 * m_WW_I5->GetMaximum() );

shared_ptr ls_comp_icn_noSLD_monly_logo = add_ILD_mark(
canvas_ls_comp_icn_noSLD_monly, 55, 1.03 * m_WW_I5->GetMaximum(), 0.1);
shared_ptr ls_comp_icn_noSLD_monly_prelim = add_prelim_mark(
canvas_ls_comp_icn_noSLD_monly, 71, 1.03 * m_WW_I5->GetMaximum(), 0.07);

string plot_name_ls_comp_icn_noSLD_monly = "./ls_comp_icn_noSLD_monly";
canvas_ls_comp_icn_noSLD_monly->Print((output_dir + plot_name_ls_comp_icn_noSLD_monly
+ ".pdf").c_str());
canvas_ls_comp_icn_noSLD_monly->Print((output_dir + plot_name_ls_comp_icn_noSLD_monly
+ ".jpg").c_str());
canvas_ls_comp_icn_noSLD_monly->Print((output_dir + plot_name_ls_comp_icn_noSLD_monly
+ ".C").c_str());
// -----

// -----

double canvas_m_m_icn_noSLD_height = 1200;
double canvas_m_m_icn_noSLD_width = 1250;

shared_ptr canvas_m_m_icn_noSLD (new TCanvas("canvas_m_m_icn_noSLD", "", 0, 0,
canvas_m_m_icn_noSLD_width, canvas_m_m_icn_noSLD_height));

m_m_WW_I5_icn_noSLD->SetLineColor(kBlue); m_m_WW_I5_icn_noSLD->SetLineStyle(1);
m_m_WW_I5_icn_noSLD->SetLineWidth(3); m_m_WW_I5_icn_noSLD-
>Scale(1.0/m_m_WW_I5_icn_noSLD->Integral());
m_m_ZZ_I5_icn_noSLD->SetLineColor(kRed); m_m_ZZ_I5_icn_noSLD->SetLineStyle(1);
m_m_ZZ_I5_icn_noSLD->SetLineWidth(3); m_m_ZZ_I5_icn_noSLD-
>Scale(1.0/m_m_ZZ_I5_icn_noSLD->Integral());
m_m_ZZ_I5_icn_noSLD->SetTitle("");
m_m_ZZ_I5_icn_noSLD->Draw("box");
m_m_WW_I5_icn_noSLD->Draw("box same");

unique_ptr leg_m_m_icn_noSLD (new TLegend(0.6, 0.5, 0.97, 0.8));
leg_m_m_icn_noSLD->SetHeader("full icn_noSLDonstr., IDR-L");
leg_m_m_icn_noSLD->Draw();

// double m_m_icn_noSLD_old_left_margin = canvas_m_m_icn_noSLD->GetLeftMargin();
// adjust_canvas_horizontal_to_square_pad(canvas_m_m_icn_noSLD);
// stack_m_m_icn_noSLD->GetYaxis()->SetTitleOffset( stack_m_m_icn_noSLD->GetYaxis()-
>GetTitleOffset() * canvas_m_m_icn_noSLD-
>GetLeftMargin()/m_m_icn_noSLD_old_left_margin );
// stack_m_m_icn_noSLD->SetMaximum( 1.15 * m_WW_I5->GetMaximum() );

```

```

shared_ptr m_m_icn_noSLD_logo = add_ILD_mark( canvas_m_m_icn_noSLD, 40, 120, 0.1);
shared_ptr m_m_icn_noSLD_prelim = add_prelim_mark( canvas_m_m_icn_noSLD, 0, 0, 0.07);

string plot_name_m_m_icn_noSLD = "./m_m_icn_noSLD";
canvas_m_m_icn_noSLD->Print((output_dir + plot_name_m_m_icn_noSLD + ".pdf").c_str());
canvas_m_m_icn_noSLD->Print((output_dir + plot_name_m_m_icn_noSLD + ".jpg").c_str());
canvas_m_m_icn_noSLD->Print((output_dir + plot_name_m_m_icn_noSLD + ".C").c_str());
// -----

// -----
// -----
// -----

for ( auto & deletable: deletables ) { delete deletable; }
for ( auto & closable: closables ) { closable->Close(); }

cout << " Done! Time for some seg faults! " << endl;
}

```