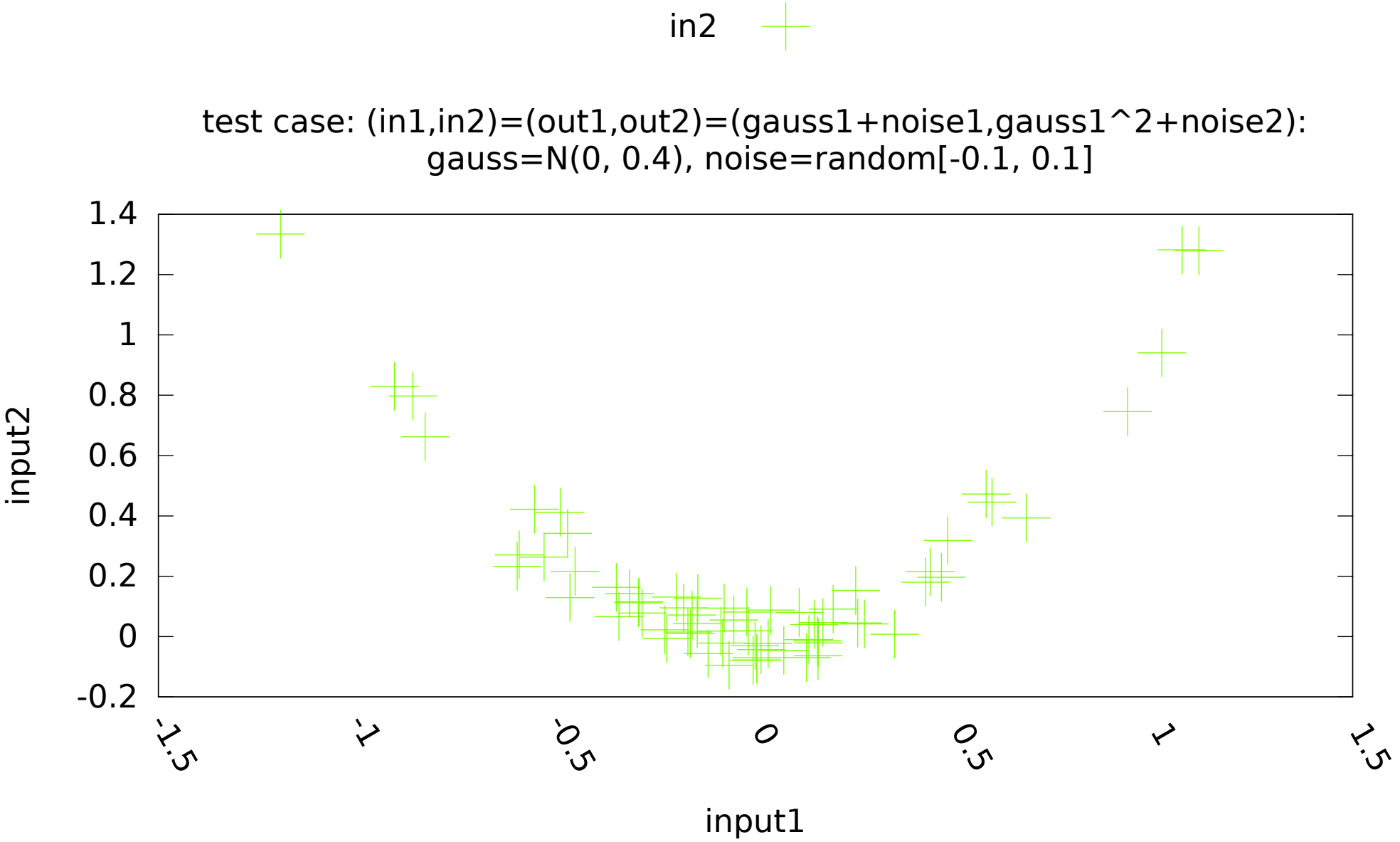
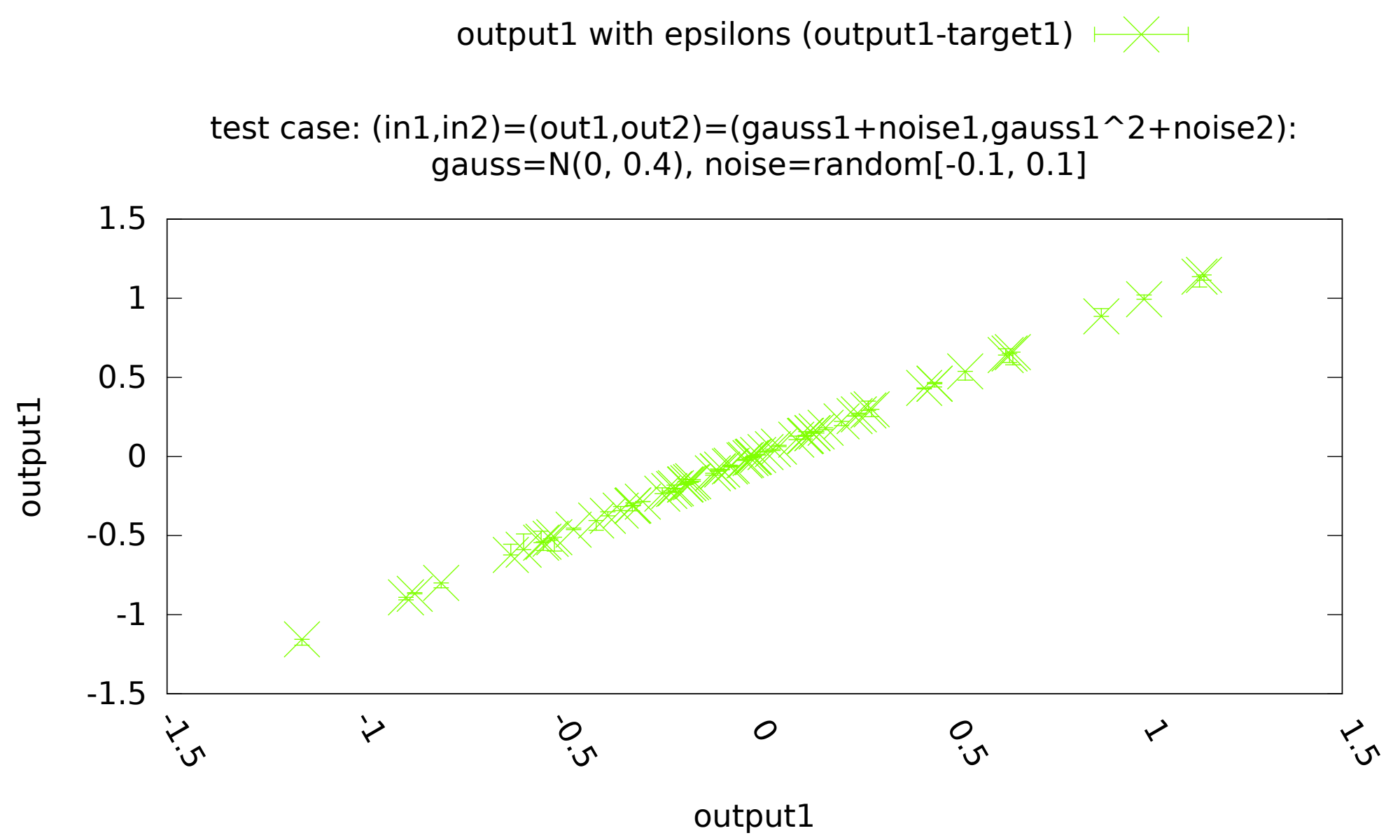
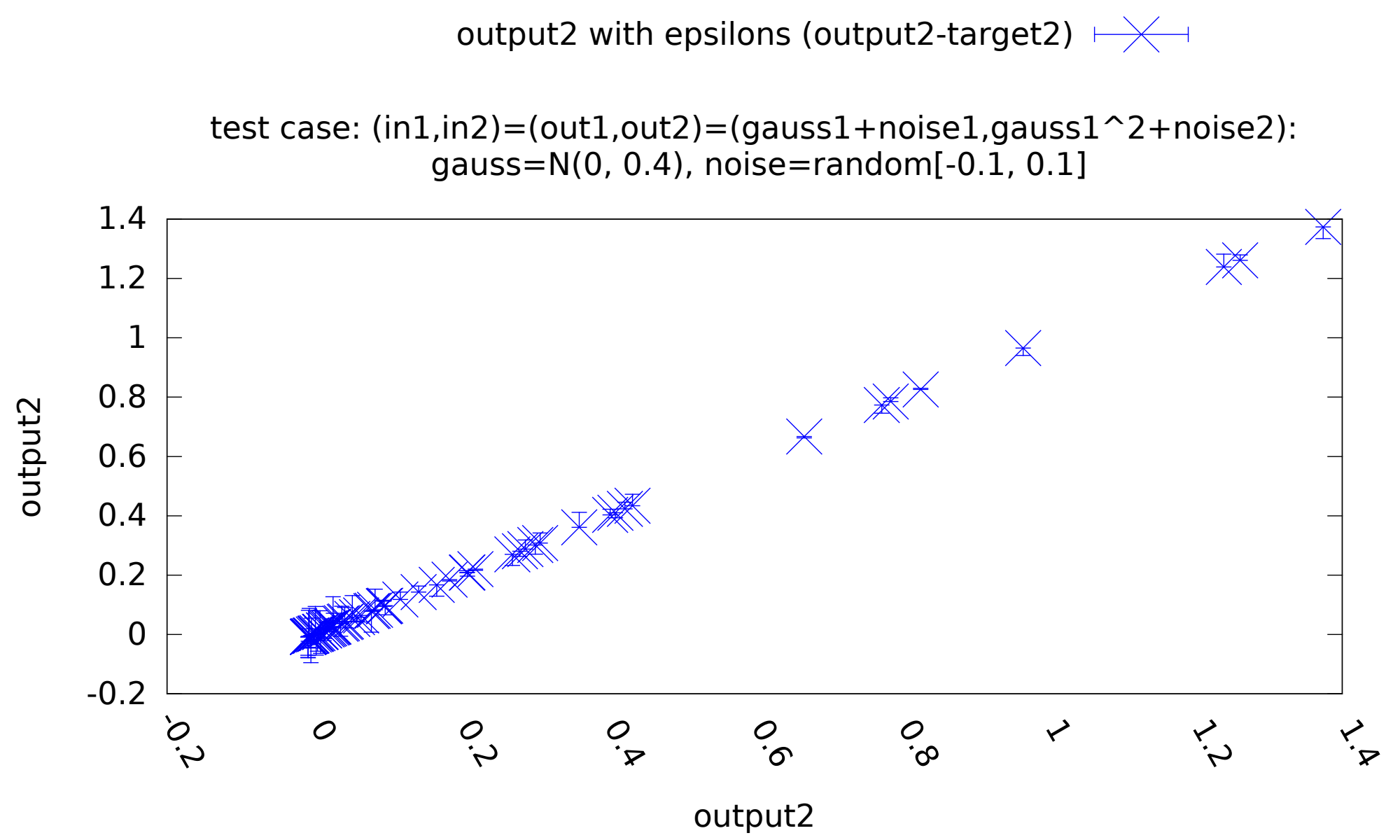
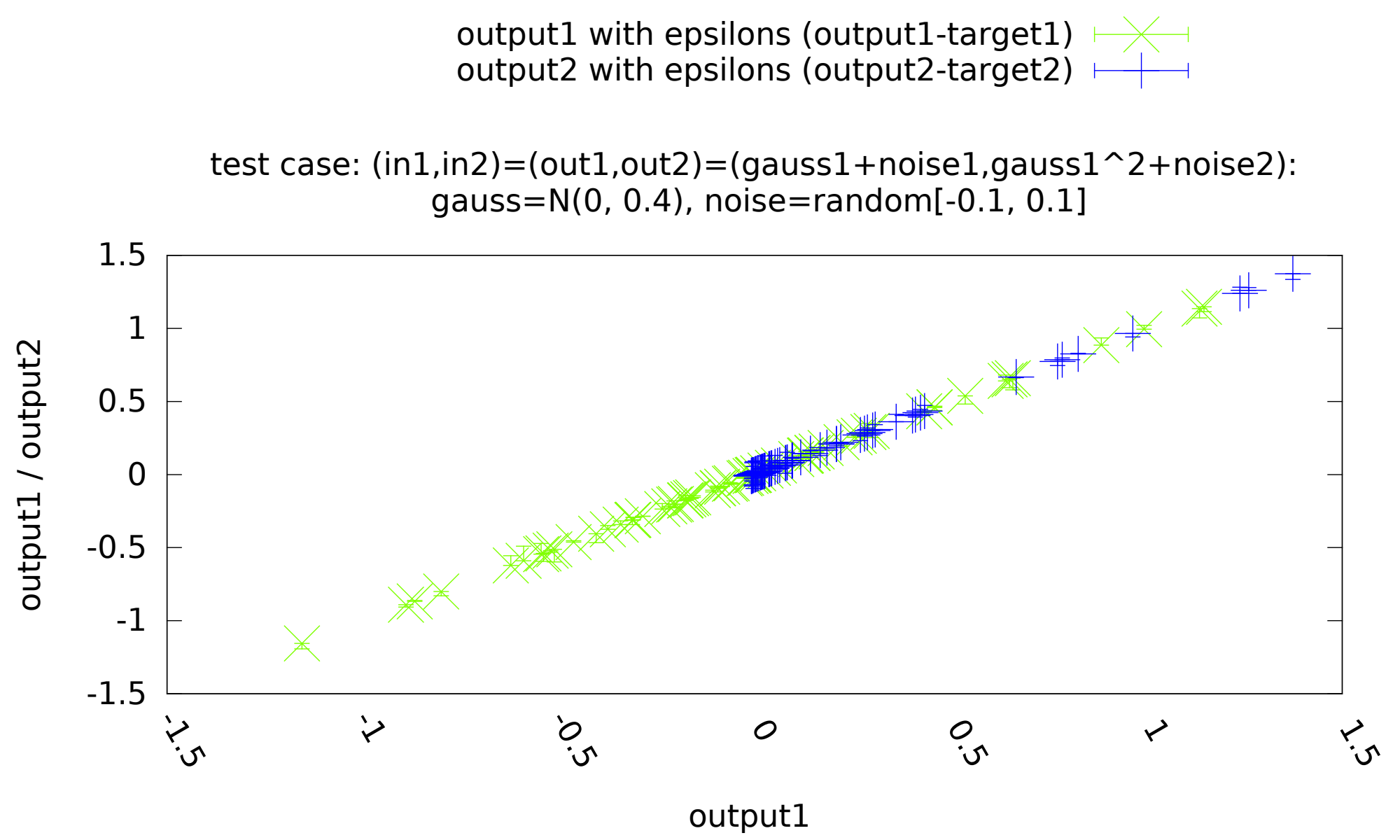
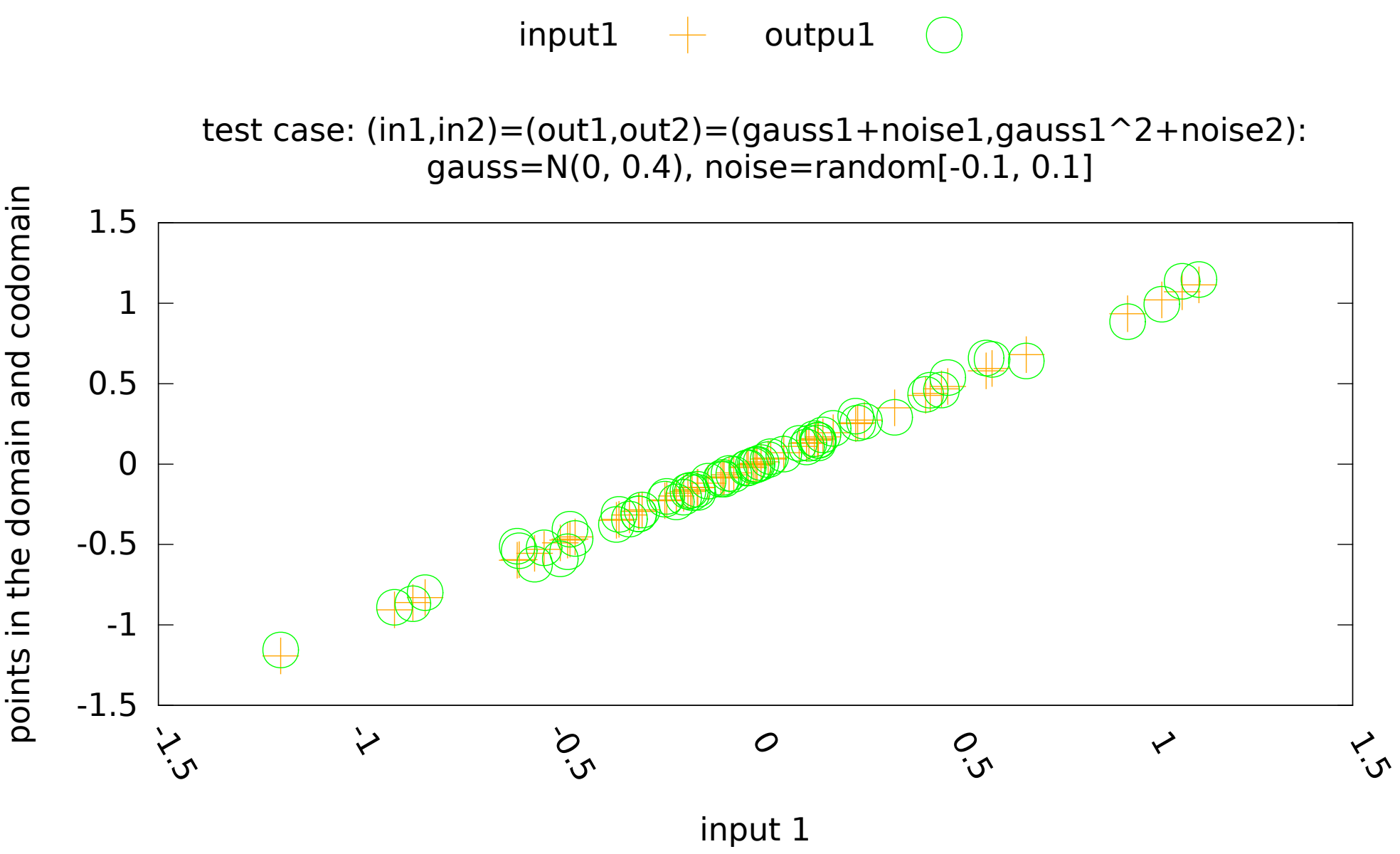
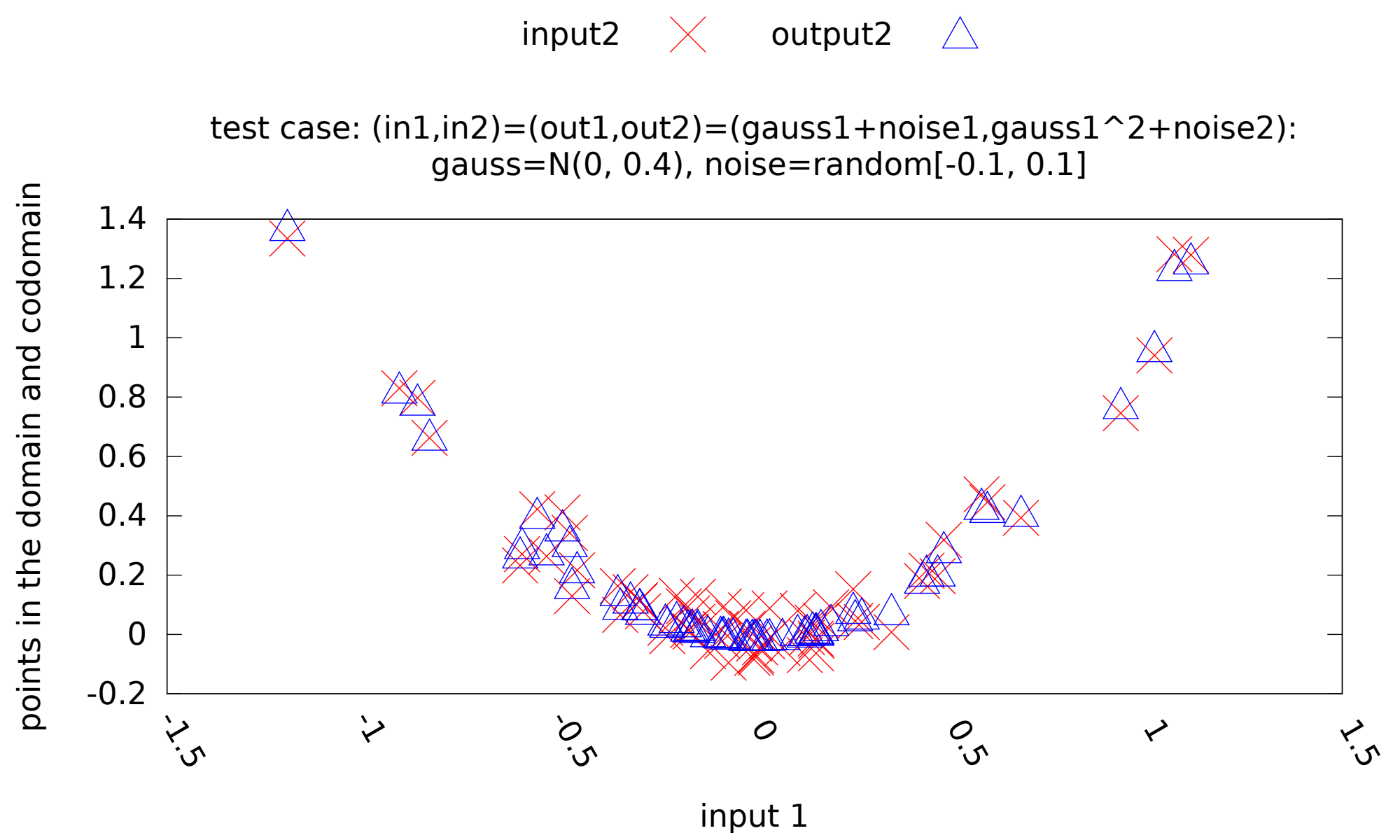

in2
test case: (in1,in2)=(out1,out2)=(gauss1+noise1,gauss1^2+noise2):
gauss=N(0, 0.4), noise=random[-0.1, 0.1]
input2
input1
1.4
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
-1.5
-1
-0.5
0
0.5
1
1.5

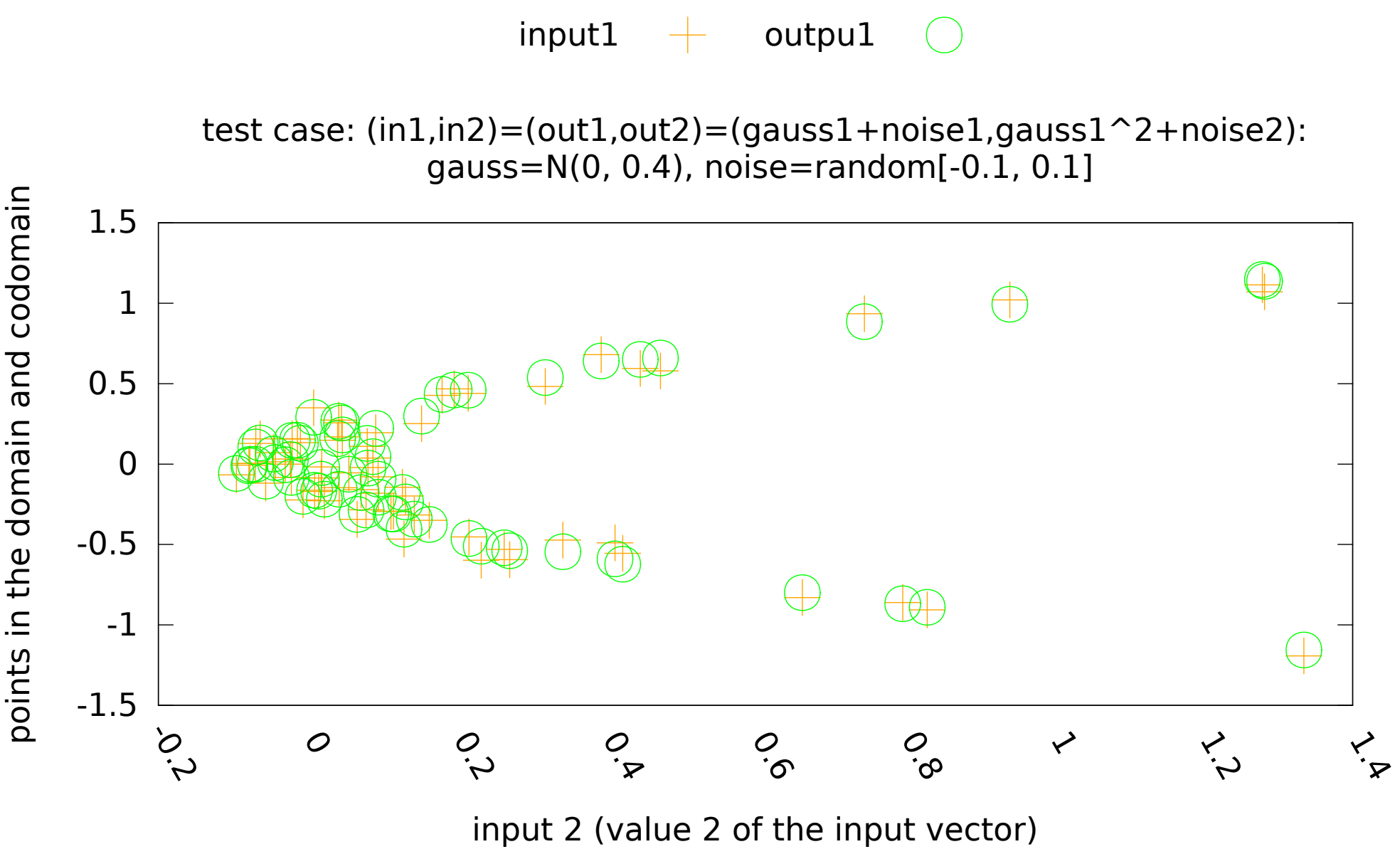

output1 with epsilons (output1-target1)
test case: (in1,in2)=(out1,out2)=(gauss1+noise1,gauss1^2+noise2):
gauss=N(0, 0.4), noise=random[-0.1, 0.1]
output1
output1
1.5
1
0.5
0
-0.5
-1
-1.5
-1.5
-1
-0.5
0
0.5
1
1.5

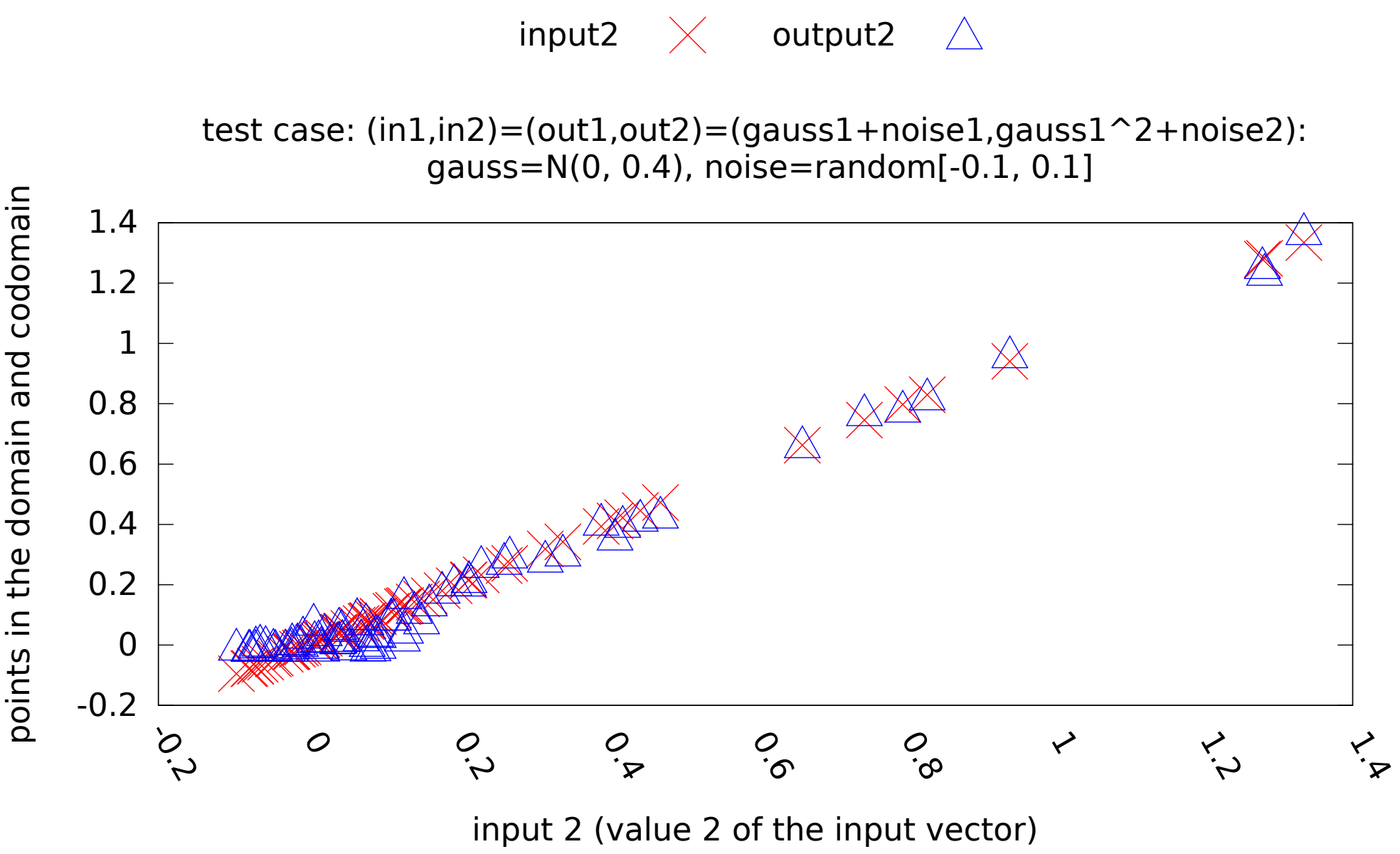

output2 with epsilons (output2-target2)
test case: (in1,in2)=(out1,out2)=(gauss1+noise1,gauss1^2+noise2):
gauss=N(0, 0.4), noise=random[-0.1, 0.1]
output2
output2
1.4
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
-0.2
0
0.2
0.4
0.6
0.8
1
1.2
1.4

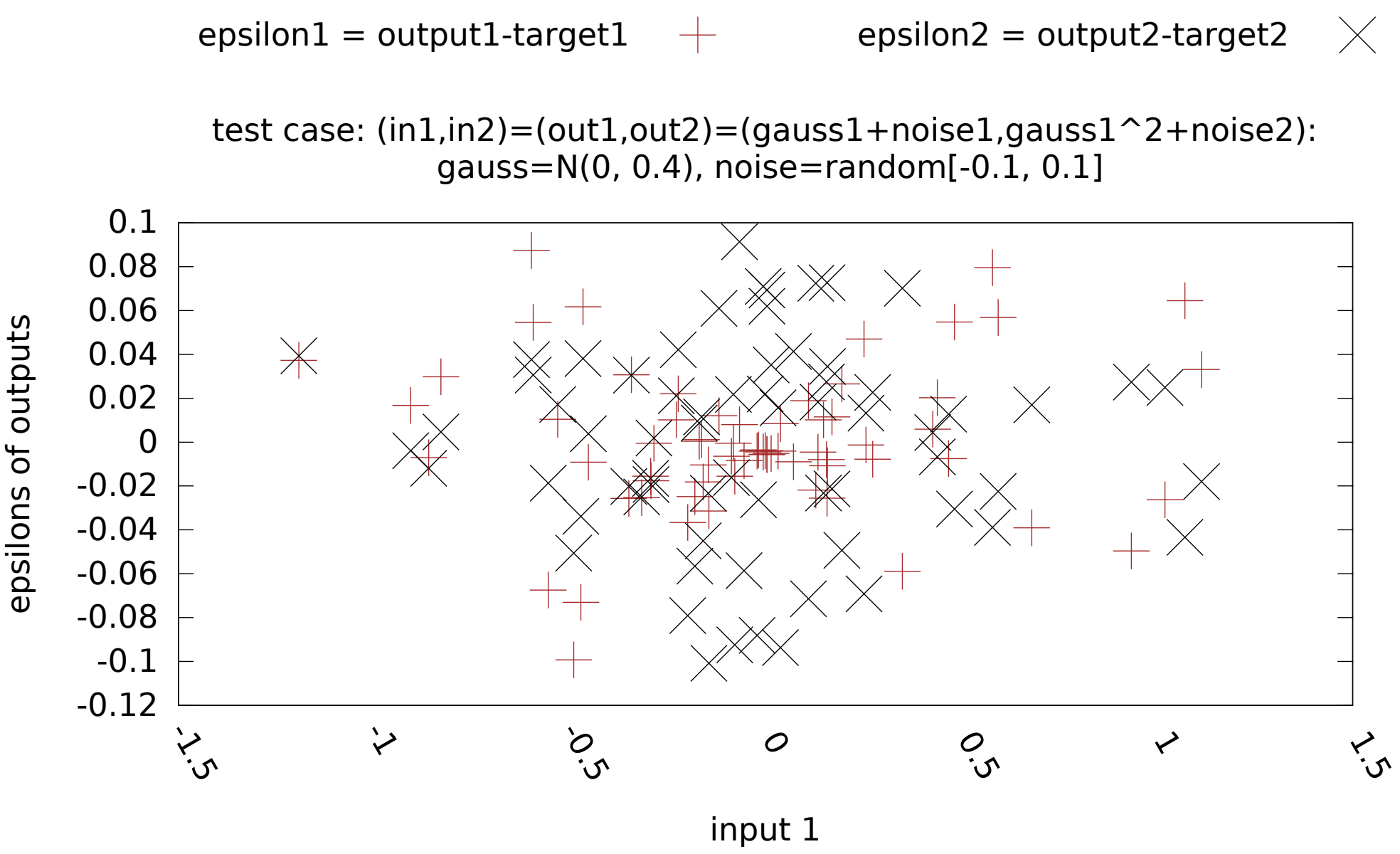

output1 with epsilons (output1-target1)
output2 with epsilons (output2-target2)
test case: (in1,in2)=(out1,out2)=(gauss1+noise1,gauss1^2+noise2):
gauss=N(0, 0.4), noise=random[-0.1, 0.1]
output1 / output2
1.5
1
0.5
0
-0.5
-1
-1.5
-1.5
-1
-0.5
0
0.5
1
1.5
output1

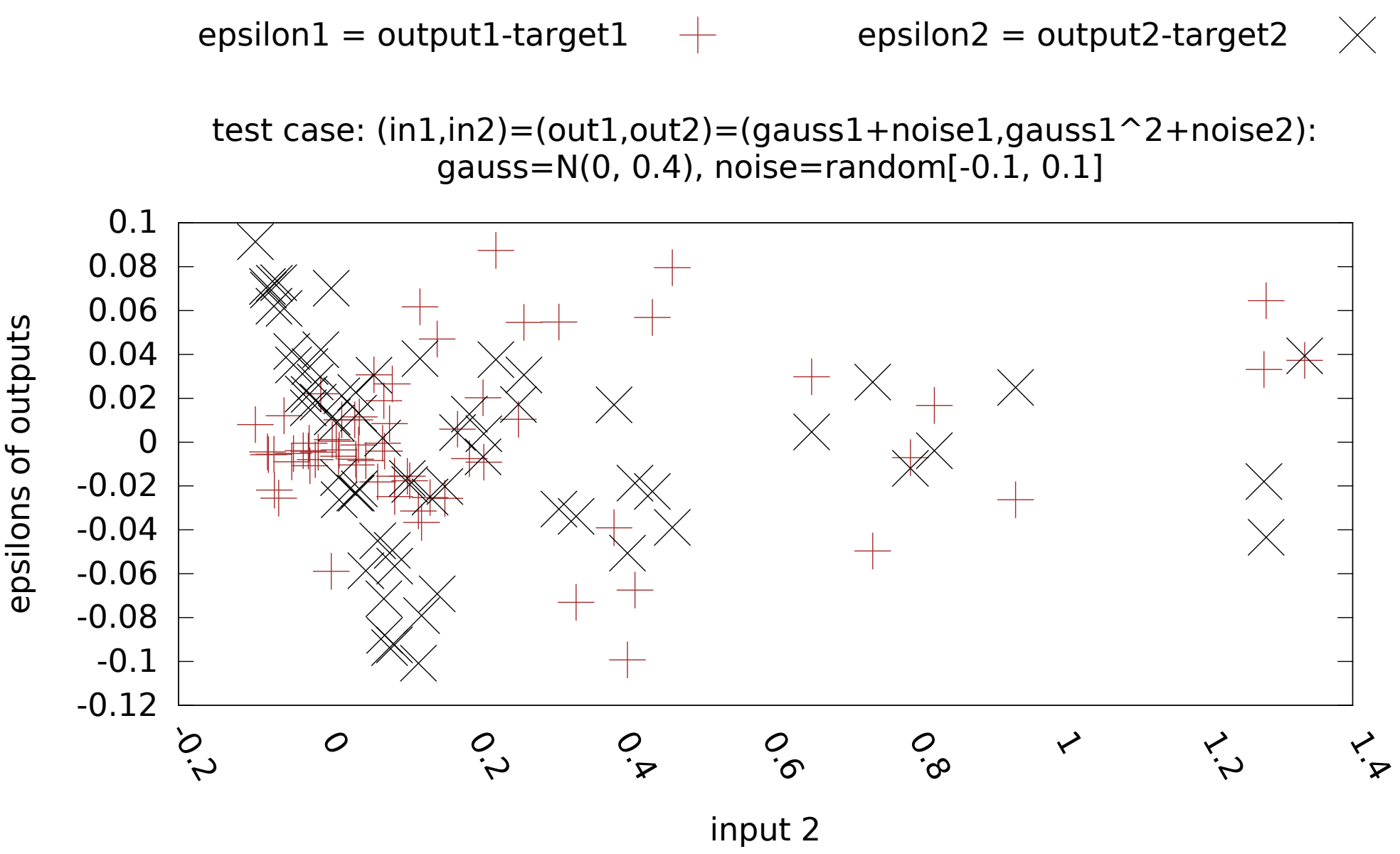

input1
outpu1
test case: (in1,in2)=(out1,out2)=(gauss1+noise1,gauss1^2+noise2):
gauss=N(0, 0.4), noise=random[-0.1, 0.1]
points in the domain and codomain
1.5
1
0.5
0
-0.5
-1
-1.5
-1.5
-1
-0.5
0
0.5
1
1.5
input 1

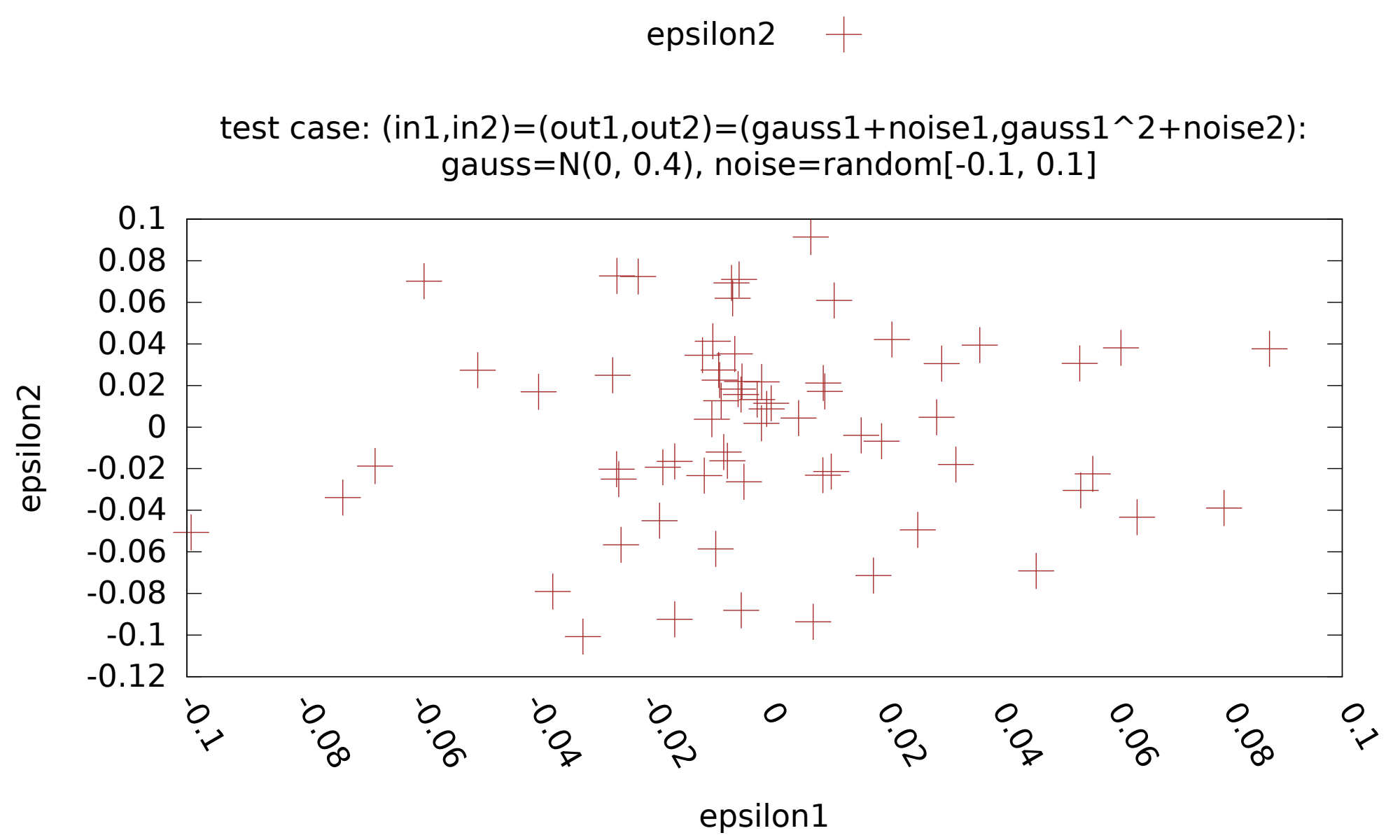

input2
output2
test case: (in1,in2)=(out1,out2)=(gauss1+noise1,gauss1^2+noise2):
gauss=N(0, 0.4), noise=random[-0.1, 0.1]
points in the domain and codomain
1.4
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
-1.5
-1
-0.5
0
0.5
1
1.5
input 1

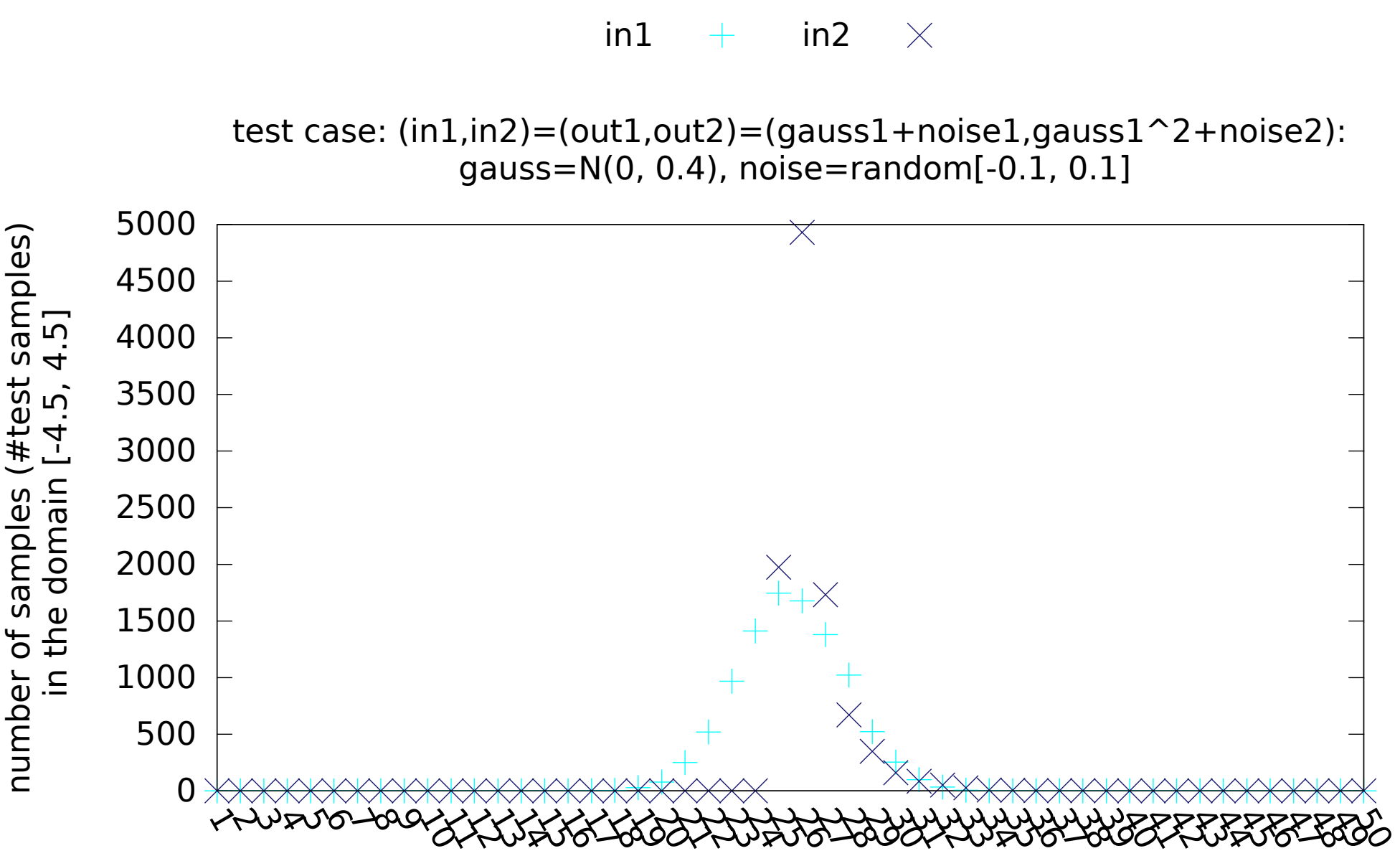

input1
outpu1
test case: (in1,in2)=(out1,out2)=(gauss1+noise1,gauss1^2+noise2):
gauss=N(0, 0.4), noise=random[-0.1, 0.1]
points in the domain and codomain
1.5
1
0.5
0
-0.5
-1
-1.5
-0.2
0
0.2
0.4
0.6
0.8
1
1.2
1.4
input 2 (value 2 of the input vector)

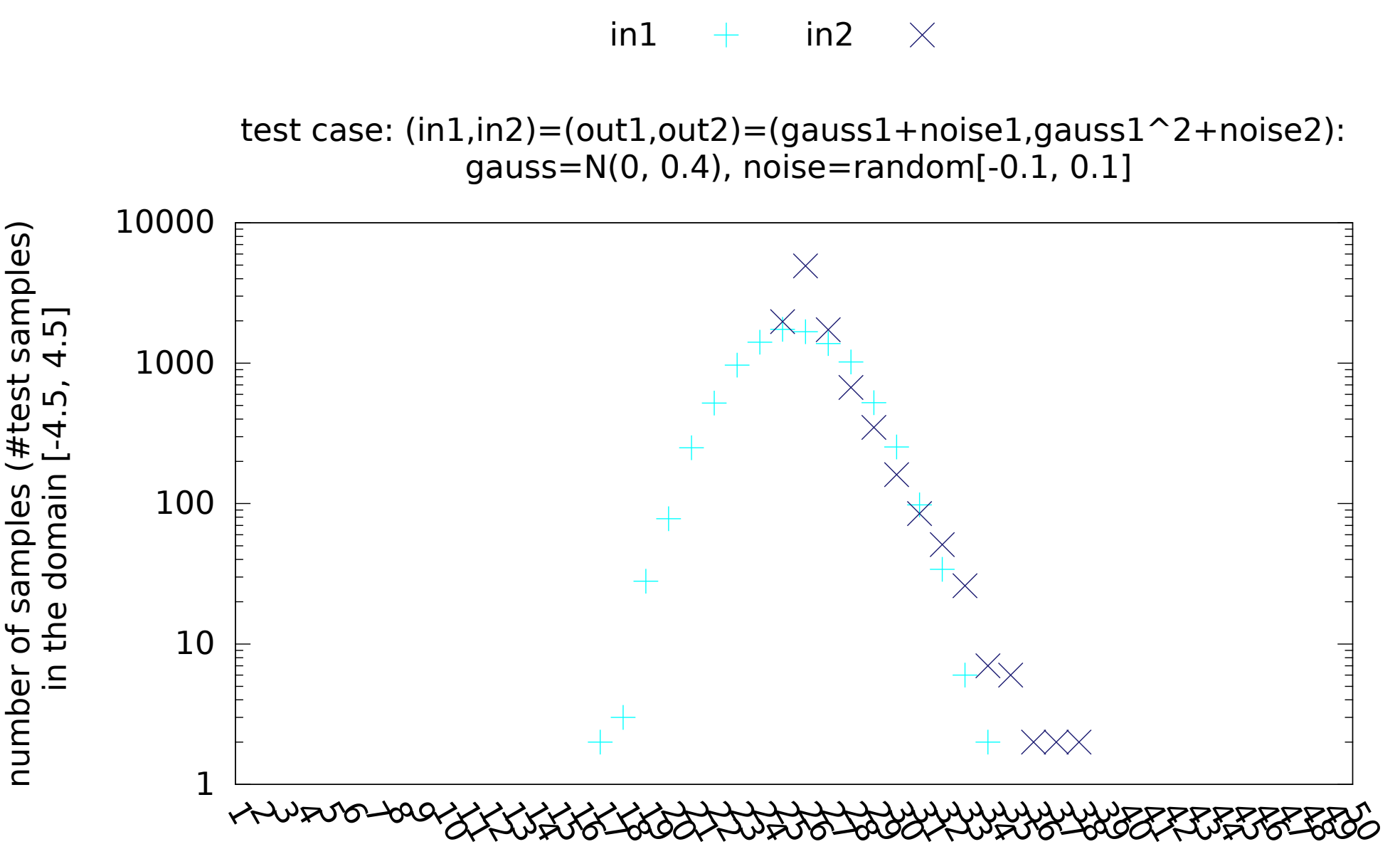

input2
output2
test case: (in1,in2)=(out1,out2)=(gauss1+noise1,gauss1^2+noise2):
gauss=N(0, 0.4), noise=random[-0.1, 0.1]
points in the domain and codomain
1.4
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
-0.2
0
0.2
0.4
0.6
0.8
1
1.2
1.4
input 2 (value 2 of the input vector)

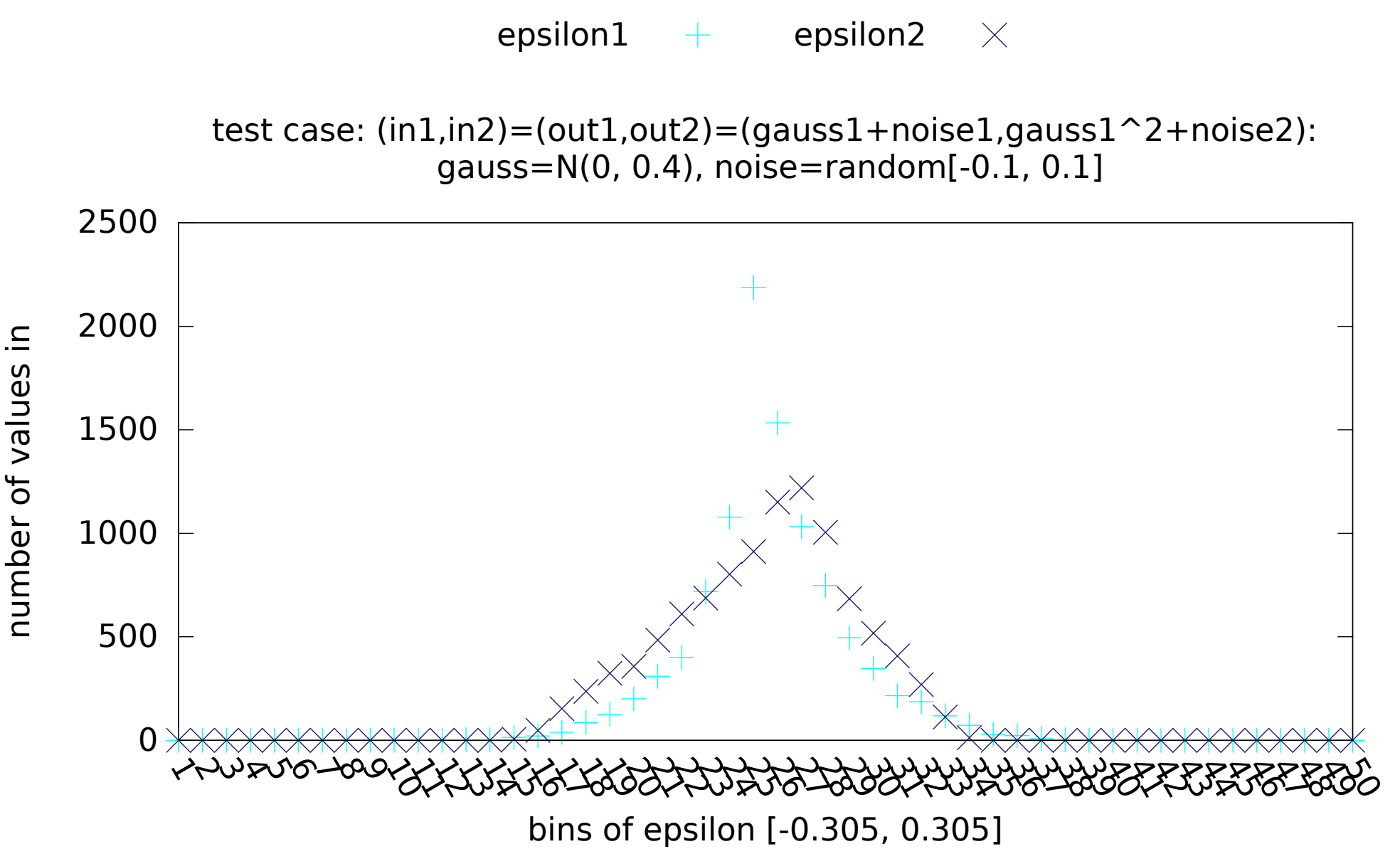

epsilon1 = output1-target1
epsilon2 = output2-target2
test case: (in1,in2)=(out1,out2)=(gauss1+noise1,gauss1^2+noise2):
gauss=N(0, 0.4), noise=random[-0.1, 0.1]
epsilons of outputs
0.1
0.08
0.06
0.04
0.02
0
-0.02
-0.04
-0.06
-0.08
-0.1
-0.12
-1.5
-1
-0.5
0
0.5
1
1.5
input 1


epsilon1 = output1-target1
epsilon2 = output2-target2
test case: (in1,in2)=(out1,out2)=(gauss1+noise1,gauss1^2+noise2):
gauss=N(0, 0.4), noise=random[-0.1, 0.1]
epsilons of outputs
0.1
0.08
0.06
0.04
0.02
0
-0.02
-0.04
-0.06
-0.08
-0.1
-0.12
-0.2
0
0.2
0.4
0.6
0.8
1
1.2
1.4
input 2


epsilon2
test case: (in1,in2)=(out1,out2)=(gauss1+noise1,gauss1^2+noise2):
gauss=N(0, 0.4), noise=random[-0.1, 0.1]
epsilon2
0.1
0.08
0.06
0.04
0.02
0
-0.02
-0.04
-0.06
-0.08
-0.1
-0.12
-0.1
-0.08
-0.06
-0.04
-0.02
0
0.02
0.04
0.06
0.08
0.1
epsilon1


in1
in2
test case: (in1,in2)=(out1,out2)=(gauss1+noise1,gauss1^2+noise2):
gauss=N(0, 0.4), noise=random[-0.1, 0.1]
number of samples (#test samples)
in the domain [-4.5, 4.5]
5000
4500
4000
3500
3000
2500
2000
1500
1000
500
0


in1
in2
test case: (in1,in2)=(out1,out2)=(gauss1+noise1,gauss1^2+noise2):
gauss=N(0, 0.4), noise=random[-0.1, 0.1]
number of samples (#test samples)
in the domain [-4.5, 4.5]
10000
1000
100
10
1


epsilon1
epsilon2
test case: (in1,in2)=(out1,out2)=(gauss1+noise1,gauss1^2+noise2):
gauss=N(0, 0.4), noise=random[-0.1, 0.1]
number of values in
2500
2000
1500
1000
500
0
bins of epsilon [-0.305, 0.305]