Appendix A Tables and graphs

Table A1 Finite sample performance of the technical efficiency estimates. σ_{u0} is lower than σ_{v0}, i.e., $\lambda_0 = 0.2$.

		Persistent TE			Transient TE			Overall TE		
n	t	R Bias	Corr	5th, 50th and 95th percentiles	R Bias	Corr	5th, 50th and 95th percentiles	R Bias	Corr	5th, 50th and 95th percentiles
S13: $\lambda_0 = 0.2,\ \lambda = 0.2,\ \Lambda = 1$ ($\sigma_{u0} = 0.04\ \sigma_u = 0.04,\ \sigma_{v0} = 0.2,\ \sigma_v = 0.2$)										
100	10	-0.037	0.117	(0.006; -0.040; -0.062)	-0.041	0.114	(0.001; -0.045; -0.066)	-0.076	0.118	(-0.022; -0.079; -0.116)
100	20	-0.038	0.115	(0.003; -0.044; -0.064)	-0.039	0.116	(0.003; -0.042; -0.065)	-0.076	0.115	(-0.022; -0.079; -0.116)
100	40	-0.039	0.117	(0.003; -0.044; -0.064)	-0.033	0.118	(0.010; -0.038; -0.060)	-0.071	0.115	(-0.018; -0.076; -0.110)
500	10	-0.043	0.114	(-0.004; -0.049; -0.069)	-0.033	0.114	(0.011; -0.038; -0.060)	-0.075	0.116	(-0.021; -0.079; -0.113)
500	20	-0.043	0.115	(-0.001; -0.049; -0.068)	-0.028	0.116	(0.015; -0.033; -0.056)	-0.07	0.113	(-0.016; -0.074; -0.110)
500	40	-0.045	0.119	(-0.003; -0.047; -0.070)	-0.022	0.118	(0.022; -0.027; -0.049)	-0.066	0.112	(-0.011; -0.069; -0.107)
1000	10	-0.042	0.115	(-0.001; -0.047; -0.067)	-0.029	0.114	(0.015; -0.033; -0.055)	-0.069	0.115	(-0.015; -0.073; -0.110)
1000	20	-0.041	0.116	(-0.000; -0.046; -0.068)	-0.023	0.117	(0.021; -0.028; -0.051)	-0.063	0.113	(-0.008; -0.068; -0.105)
1000	40	-0.042	0.118	(-0.001; -0.046; -0.068)	-0.018	0.119	(0.027; -0.023; -0.046)	-0.06	0.111	(-0.004; -0.063; -0.101)
S14: $\lambda_0 = 0.2,\ \lambda = 1,\ \Lambda = 0.2$ ($\sigma_{u0} = 0.04\ \sigma_u = 0.02,\ \sigma_{v0} = 0.2,\ \sigma_v = 0.2$)										
100	10	-0.036	0.11	(0.006; -0.041; -0.063)	0.038	0.502	(0.245; 0.015; -0.086)	0	0.45	(0.202; -0.020; -0.124)
100	20	-0.039	0.117	(0.004; -0.044; -0.063)	0.027	0.515	(0.224; 0.005; -0.092)	-0.013	0.467	(0.175; -0.034; -0.134)
100	40	-0.039	0.112	(0.004; -0.043; -0.064)	0.018	0.521	(0.207; -0.004; -0.096)	-0.021	0.482	(0.162; -0.042; -0.138)
500	10	-0.043	0.113	(-0.002; -0.047; -0.069)	0.019	0.503	(0.209; -0.004; -0.097)	-0.025	0.465	(0.165; -0.043; -0.141)
500	20	-0.045	0.118	(-0.006; -0.050; -0.071)	0.016	0.515	(0.202; -0.004; -0.097)	-0.03	0.475	(0.152; -0.048; -0.147)
500	40	-0.044	0.117	(-0.004; -0.046; -0.069)	0.013	0.521	(0.190; -0.004; -0.099)	-0.031	0.482	(0.149; -0.050; -0.145)
1000	10	-0.041	0.113	(-0.000; -0.045; -0.067)	0.016	0.504	(0.207; -0.005; -0.099)	-0.026	0.474	(0.159; -0.041; -0.142)
1000	20	-0.042	0.115	(-0.002; -0.046; -0.067)	0.014	0.515	(0.193; -0.008; -0.102)	-0.029	0.483	(0.152; -0.049; -0.145)
1000	40	-0.041	0.119	(-0.001; -0.047; -0.067)	0.013	0.521	(0.198; -0.008; -0.099)	-0.029	0.488	(0.153; -0.048; -0.146)
S9: $\lambda_0 = 0.2,\ \lambda = 1,\ \Lambda = 1$ ($\sigma_{u0} = 0.04\ \sigma_u = 0.04,\ \sigma_{v0} = 0.2,\ \sigma_v = 0.04$)										
100	10	-0.037	0.117	(0.007; -0.041; -0.062)	0.006	0.504	(0.044; 0.002; -0.019)	-0.032	0.243	(0.022; -0.035; -0.070)

100	20	-0.038	0.122	(0.004; -0.042; -0.063)	0.003	0.519	(0.039; 0.000; -0.020)	-0.035	0.265	(0.017; -0.039; -0.073)
100	40	-0.04	0.12	(0.002; -0.043; -0.064)	0.002	0.526	(0.037; -0.002; -0.021)	-0.038	0.276	(0.010; -0.042; -0.076)
500	10	-0.042	0.117	(-0.002; -0.047; -0.066)	0.002	0.505	(0.038; -0.001; -0.021)	-0.04	0.256	(0.012; -0.043; -0.078)
500	20	-0.043	0.117	(-0.003; -0.047; -0.068)	0.001	0.519	(0.036; -0.002; -0.021)	-0.042	0.265	(0.008; -0.046; -0.080)
500	40	-0.044	0.119	(-0.004; -0.048; -0.069)	0.001	0.526	(0.035; -0.002; -0.021)	-0.043	0.269	(0.005; -0.047; -0.080)
1000	10	-0.042	0.119	(-0.003; -0.047; -0.067)	0.002	0.506	(0.037; -0.002; -0.021)	-0.041	0.263	(0.008; -0.044; -0.078)
1000	20	-0.042	0.117	(-0.001; -0.045; -0.067)	0.001	0.519	(0.035; -0.002; -0.022)	-0.041	0.273	(0.009; -0.045; -0.079)
1000	40	-0.04	0.118	(-0.000; -0.045; -0.066)	0.001	0.526	(0.035; -0.002; -0.021)	-0.039	0.278	(0.010; -0.043; -0.076)
S10: $\lambda_0 = 0.2,\ \lambda = 5,\ \Lambda = 0.2$ ($\sigma_{u0} = 0.04\ \sigma_u = 0.2,\ \sigma_{v0} = 0.2,\ \sigma_v = 0.04$)										
100	10	-0.039	0.11	(0.004; -0.042; -0.065)	0.01	0.92	(0.044; 0.016; -0.036)	-0.029	0.884	(0.005; -0.026; -0.079)
100	20	-0.04	0.112	(0.002; -0.045; -0.065)	0.005	0.935	(0.027; 0.012; -0.031)	-0.035	0.897	(-0.011; -0.029; -0.079)
100	40	-0.041	0.116	(-0.000; -0.045; -0.067)	0.003	0.942	(0.021; 0.009; -0.029)	-0.038	0.903	(-0.014; -0.033; -0.079)
500	10	-0.044	0.117	(-0.003; -0.049; -0.070)	0.009	0.921	(0.039; 0.013; -0.034)	-0.036	0.882	(-0.002; -0.030; -0.085)
500	20	-0.044	0.117	(-0.004; -0.050; -0.069)	0.005	0.936	(0.024; 0.010; -0.031)	-0.04	0.898	(-0.012; -0.036; -0.081)
500	40	-0.044	0.118	(-0.002; -0.050; -0.069)	0.003	0.943	(0.021; 0.008; -0.028)	-0.041	0.904	(-0.021; -0.036; -0.080)
1000	10	-0.042	0.119	(-0.002; -0.046; -0.067)	0.009	0.921	(0.039; 0.015; -0.035)	-0.033	0.887	(-0.001; -0.028; -0.081)
1000	20	-0.042	0.119	(-0.000; -0.046; -0.067)	0.005	0.936	(0.027; 0.010; -0.032)	-0.037	0.902	(-0.008; -0.031; -0.078)
1000	40	-0.041	0.118	(0.000; -0.046; -0.067)	0.003	0.943	(0.021; 0.009; -0.029)	-0.038	0.909	(-0.015; -0.032; -0.079)

Table A2 Finite sample performance of the technical efficiency estimates. σ_{u0} equals σ_{v0}, i.e., $\lambda_0 = 1$.

		Persistent TE			Transient TE			Overall TE		
n	t	R Bias	Corr	5th, 50th and 95th percentiles	R Bias	Corr	5th, 50th and 95th percentiles	R Bias	Corr	5th, 50th and 95th percentiles
s5: $\lambda_0 = 1,\ \lambda = 0.2,\ \Lambda = 1$ ($\sigma_{u0} = 0.04\ \sigma_u = 0.04,\ \sigma_{v0} = 0.04,\ \sigma_v = 0.2$)										
100	10	0.018	0.307	(0.066; 0.013; -0.011)	-0.041	0.115	(-0.001; -0.046; -0.067)	-0.024	0.158	(0.033; -0.029; -0.066)
100	20	0.017	0.378	(0.065; 0.013; -0.011)	-0.037	0.117	(0.004; -0.042; -0.063)	-0.021	0.164	(0.038; -0.025; -0.062)
100	40	0.016	0.431	(0.062; 0.011; -0.012)	-0.035	0.118	(0.007; -0.039; -0.062)	-0.02	0.179	(0.037; -0.023; -0.062)
500	10	0.018	0.308	(0.065; 0.013; -0.011)	-0.032	0.117	(0.010; -0.036; -0.059)	-0.015	0.165	(0.044; -0.020; -0.059)
500	20	0.016	0.381	(0.062; 0.011; -0.013)	-0.028	0.117	(0.015; -0.032; -0.054)	-0.013	0.187	(0.047; -0.018; -0.056)
500	40	0.01	0.434	(0.053; 0.006; -0.016)	-0.023	0.119	(0.021; -0.028; -0.050)	-0.013	0.238	(0.044; -0.017; -0.055)
1000	10	0.018	0.31	(0.066; 0.013; -0.011)	-0.027	0.117	(0.016; -0.032; -0.054)	-0.009	0.169	(0.050; -0.014; -0.053)
1000	20	0.015	0.379	(0.061; 0.010; -0.013)	-0.023	0.119	(0.021; -0.028; -0.049)	-0.008	0.2	(0.051; -0.012; -0.052)
1000	40	0.008	0.437	(0.049; 0.004; -0.018)	-0.018	0.119	(0.027; -0.023; -0.046)	-0.01	0.263	(0.047; -0.014; -0.053)
s15: $\lambda_0 = 1,\ \lambda = 0.2,\ \Lambda = 5$ ($\sigma_{u0} = 0.2\ \sigma_u = 0.04,\ \sigma_{v0} = 0.2,\ \sigma_v = 0.2$)										
100	10	0.079	0.502	(0.323; 0.051; -0.057)	-0.042	0.112	(-0.002; -0.046; -0.068)	0.034	0.388	(0.261; 0.008; -0.104)
100	20	0.075	0.517	(0.316; 0.050; -0.061)	-0.038	0.116	(0.004; -0.043; -0.064)	0.033	0.413	(0.260; 0.008; -0.105)
100	40	0.071	0.518	(0.310; 0.047; -0.061)	-0.034	0.117	(0.007; -0.039; -0.060)	0.034	0.428	(0.258; 0.010; -0.101)
500	10	0.05	0.51	(0.264; 0.026; -0.077)	-0.035	0.113	(0.008; -0.038; -0.061)	0.013	0.454	(0.221; -0.011; -0.118)
500	20	0.045	0.517	(0.258; 0.019; -0.081)	-0.028	0.117	(0.016; -0.032; -0.054)	0.016	0.474	(0.225; -0.007; -0.114)
500	40	0.043	0.521	(0.249; 0.020; -0.082)	-0.022	0.118	(0.022; -0.027; -0.050)	0.019	0.489	(0.226; -0.003; -0.109)
1000	10	0.035	0.508	(0.242; 0.013; -0.087)	-0.028	0.114	(0.016; -0.033; -0.055)	0.006	0.473	(0.203; -0.017; -0.119)
1000	20	0.034	0.517	(0.239; 0.012; -0.088)	-0.023	0.116	(0.021; -0.028; -0.050)	0.01	0.49	(0.210; -0.011; -0.115)
1000	40	0.03	0.523	(0.226; 0.008; -0.088)	-0.018	0.118	(0.026; -0.023; -0.046)	0.011	0.501	(0.207; -0.009; -0.113)
s6: $\lambda_0 = 1,\ \lambda = 1,\ \Lambda = 0.2$ ($\sigma_{u0} = 0.04\ \sigma_u = 0.2,\ \sigma_{v0} = 0.04,\ \sigma_v = 0.2$)										

100	10	0.018	0.278	(0.066; 0.013; -0.011)	0.033	0.519	(0.231; 0.010; -0.086)	0.051	0.524	(0.254; 0.028; -0.077)
100	20	0.018	0.351	(0.065; 0.013; -0.011)	0.024	0.522	(0.215; 0.002; -0.094)	0.042	0.525	(0.243; 0.020; -0.085)
100	40	0.016	0.412	(0.063; 0.012; -0.012)	0.019	0.524	(0.201; -0.005; -0.097)	0.036	0.524	(0.231; 0.017; -0.088)
500	10	0.018	0.278	(0.065; 0.013; -0.011)	0.017	0.519	(0.204; -0.004; -0.098)	0.035	0.524	(0.228; 0.012; -0.088)
500	20	0.016	0.349	(0.063; 0.012; -0.012)	0.015	0.522	(0.197; -0.006; -0.099)	0.031	0.524	(0.225; 0.012; -0.091)
500	40	0.012	0.415	(0.055; 0.007; -0.015)	0.014	0.524	(0.199; -0.005; -0.099)	0.025	0.525	(0.214; 0.006; -0.094)
1000	10	0.019	0.278	(0.066; 0.014; -0.010)	0.014	0.52	(0.197; -0.005; -0.102)	0.033	0.524	(0.221; 0.015; -0.088)
1000	20	0.017	0.347	(0.063; 0.012; -0.012)	0.013	0.522	(0.197; -0.006; -0.099)	0.03	0.524	(0.216; 0.010; -0.092)
1000	40	0.011	0.416	(0.054; 0.006; -0.016)	0.013	0.524	(0.188; -0.007; -0.097)	0.024	0.525	(0.210; 0.004; -0.094)

s1: $\lambda_0 = 1,\ \lambda = 1,\ \Lambda = 1$ ($\sigma_{u0} = 0.04\ \sigma_u = 0.04,\ \sigma_{v0} = 0.04,\ \sigma_v = 0.04$)

100	10	0.013	0.506	(0.056; 0.009; -0.013)	0.006	0.508	(0.043; 0.002; -0.018)	0.019	0.499	(0.072; 0.016; -0.021)
100	20	0.013	0.515	(0.056; 0.009; -0.013)	0.003	0.52	(0.039; 0.000; -0.020)	0.016	0.491	(0.068; 0.012; -0.023)
100	40	0.013	0.527	(0.055; 0.009; -0.013)	0.002	0.526	(0.037; -0.002; -0.021)	0.014	0.491	(0.064; 0.011; -0.025)
500	10	0.008	0.506	(0.047; 0.004; -0.017)	0.002	0.508	(0.038; -0.002; -0.021)	0.01	0.505	(0.058; 0.006; -0.028)
500	20	0.007	0.522	(0.045; 0.003; -0.017)	0.001	0.52	(0.036; -0.002; -0.022)	0.008	0.507	(0.057; 0.006; -0.029)
500	40	0.007	0.527	(0.044; 0.003; -0.018)	0.001	0.526	(0.036; -0.003; -0.021)	0.008	0.508	(0.056; 0.004; -0.029)
1000	10	0.006	0.507	(0.043; 0.002; -0.019)	0.002	0.508	(0.037; -0.002; -0.022)	0.007	0.513	(0.054; 0.003; -0.030)
1000	20	0.005	0.52	(0.042; 0.002; -0.019)	0.001	0.52	(0.036; -0.002; -0.021)	0.006	0.514	(0.054; 0.003; -0.030)
1000	40	0.005	0.526	(0.041; 0.001; -0.019)	0.001	0.526	(0.035; -0.003; -0.022)	0.006	0.515	(0.052; 0.002; -0.031)

s16: $\lambda_0 = 1,\ \lambda = 1,\ \Lambda = 1$ ($\sigma_{u0} = 0.2\ \sigma_u = 0.2,\ \sigma_{v0} = 0.2,\ \sigma_v = 0.2$)

100	10	0.077	0.498	(0.319; 0.050; -0.061)	0.035	0.502	(0.244; 0.014; -0.086)	0.115	0.491	(0.426; 0.087; -0.096)
100	20	0.075	0.511	(0.314; 0.050; -0.061)	0.025	0.515	(0.220; 0.002; -0.094)	0.101	0.486	(0.405; 0.074; -0.105)
100	40	0.075	0.516	(0.316; 0.049; -0.061)	0.019	0.521	(0.205; -0.002; -0.094)	0.095	0.482	(0.394; 0.066; -0.112)
500	10	0.049	0.503	(0.267; 0.025; -0.080)	0.02	0.503	(0.215; -0.000; -0.099)	0.07	0.499	(0.357; 0.043; -0.126)
500	20	0.046	0.514	(0.258; 0.022; -0.080)	0.015	0.515	(0.202; -0.006; -0.102)	0.061	0.5	(0.334; 0.035; -0.133)
500	40	0.042	0.521	(0.250; 0.018; -0.083)	0.014	0.521	(0.197; -0.006; -0.101)	0.056	0.503	(0.315; 0.031; -0.133)

1000	10	0.038	0.503	(0.247; 0.015; -0.087)	0.016	0.503	(0.207; -0.005; -0.101)	0.054	0.506	(0.334; 0.031; -0.135)
1000	20	0.035	0.514	(0.236; 0.012; -0.088)	0.014	0.515	(0.193; -0.008; -0.099)	0.049	0.508	(0.321; 0.026; -0.140)
1000	40	0.034	0.52	(0.232; 0.012; -0.086)	0.013	0.521	(0.199; -0.006; -0.100)	0.048	0.509	(0.313; 0.026; -0.138)
s11: $\lambda_0 = 1,\ \lambda = 1,\ \Lambda = 5$ ($\sigma_{u0} = 0.2\ \sigma_u = 0.04,\ \sigma_{v0} = 0.2,\ \sigma_v = 0.04$)										
100	10	0.073	0.521	(0.313; 0.048; -0.061)	0.006	0.505	(0.044; 0.002; -0.018)	0.079	0.499	(0.312; 0.054; -0.061)
100	20	0.073	0.526	(0.309; 0.049; -0.061)	0.003	0.519	(0.039; 0.000; -0.020)	0.076	0.5	(0.302; 0.052; -0.063)
100	40	0.072	0.523	(0.305; 0.047; -0.061)	0.002	0.526	(0.037; -0.001; -0.021)	0.074	0.495	(0.303; 0.048; -0.065)
500	10	0.046	0.526	(0.254; 0.022; -0.079)	0.002	0.505	(0.038; -0.002; -0.021)	0.049	0.512	(0.262; 0.025; -0.083)
500	20	0.041	0.528	(0.243; 0.020; -0.082)	0.001	0.519	(0.036; -0.002; -0.021)	0.042	0.514	(0.248; 0.017; -0.085)
500	40	0.044	0.527	(0.252; 0.021; -0.080)	0.001	0.526	(0.035; -0.002; -0.021)	0.045	0.513	(0.252; 0.021; -0.085)
1000	10	0.034	0.525	(0.233; 0.012; -0.087)	0.002	0.505	(0.037; -0.002; -0.021)	0.036	0.517	(0.239; 0.014; -0.089)
1000	20	0.031	0.527	(0.228; 0.008; -0.088)	0.001	0.519	(0.036; -0.003; -0.021)	0.032	0.52	(0.230; 0.014; -0.092)
1000	40	0.032	0.527	(0.230; 0.008; -0.086)	0.001	0.526	(0.035; -0.002; -0.021)	0.033	0.518	(0.235; 0.009; -0.092)
s2: $\lambda_0 = 1,\ \lambda = 5,\ \Lambda = 0.2$ ($\sigma_{u0} = 0.04\ \sigma_u = 0.2,\ \sigma_{v0} = 0.04,\ \sigma_v = 0.04$)										
100	10	0.016	0.431	(0.062; 0.011; -0.012)	0.006	0.928	(0.030; 0.009; -0.033)	0.022	0.923	(0.050; 0.028; -0.025)
100	20	0.014	0.478	(0.058; 0.010; -0.013)	0.004	0.938	(0.026; 0.009; -0.032)	0.018	0.927	(0.045; 0.023; -0.024)
100	40	0.012	0.503	(0.055; 0.008; -0.013)	0.003	0.943	(0.023; 0.007; -0.029)	0.015	0.93	(0.038; 0.022; -0.026)
500	10	0.005	0.436	(0.043; 0.001; -0.020)	0.005	0.929	(0.027; 0.008; -0.033)	0.01	0.924	(0.036; 0.014; -0.030)
500	20	0.007	0.479	(0.046; 0.003; -0.018)	0.004	0.938	(0.026; 0.009; -0.030)	0.011	0.929	(0.035; 0.017; -0.029)
500	40	0.007	0.503	(0.044; 0.003; -0.018)	0.003	0.943	(0.021; 0.007; -0.029)	0.009	0.931	(0.031; 0.012; -0.031)
1000	10	0.001	0.433	(0.038; -0.003; -0.023)	0.005	0.929	(0.026; 0.008; -0.033)	0.005	0.925	(0.030; 0.010; -0.035)
1000	20	0.004	0.481	(0.041; -0.000; -0.020)	0.004	0.938	(0.023; 0.008; -0.030)	0.007	0.929	(0.030; 0.014; -0.033)
1000	40	0.005	0.506	(0.041; 0.001; -0.019)	0.003	0.943	(0.023; 0.009; -0.028)	0.007	0.932	(0.028; 0.012; -0.030)
s12: $\lambda_0 = 1,\ \lambda = 5,\ \Lambda = 1$ ($\sigma_{u0} = 0.2\ \sigma_u = 0.2,\ \sigma_{v0} = 0.2,\ \sigma_v = 0.04$)										
100	10	0.074	0.52	(0.311; 0.048; -0.061)	0.01	0.919	(0.039; 0.017; -0.035)	0.084	0.725	(0.266; 0.069; -0.056)
100	20	0.071	0.519	(0.309; 0.047; -0.062)	0.005	0.936	(0.028; 0.009; -0.031)	0.076	0.73	(0.252; 0.069; -0.056)

100	40	0.071	0.519	(0.308; 0.047; -0.063)	0.003	0.943	(0.025; 0.010; -0.029)	0.074	0.732	(0.240; 0.063; -0.054)
500	10	0.044	0.521	(0.249; 0.022; -0.080)	0.009	0.92	(0.038; 0.014; -0.035)	0.054	0.737	(0.213; 0.044; -0.075)
500	20	0.043	0.524	(0.247; 0.020; -0.080)	0.005	0.936	(0.025; 0.011; -0.030)	0.048	0.742	(0.200; 0.036; -0.074)
500	40	0.043	0.523	(0.250; 0.020; -0.081)	0.003	0.943	(0.022; 0.008; -0.029)	0.046	0.745	(0.198; 0.036; -0.072)
1000	10	0.03	0.522	(0.222; 0.008; -0.091)	0.009	0.92	(0.034; 0.015; -0.035)	0.039	0.745	(0.191; 0.032; -0.083)
1000	20	0.029	0.525	(0.223; 0.006; -0.089)	0.005	0.936	(0.027; 0.010; -0.030)	0.034	0.75	(0.179; 0.026; -0.081)
1000	40	0.032	0.527	(0.229; 0.009; -0.088)	0.003	0.943	(0.019; 0.008; -0.029)	0.035	0.752	(0.178; 0.024; -0.080)

Table A3 Finite sample performance of the technical efficiency estimates. $\sigma_{u0} > \sigma_{v0}$, i.e., $\lambda_0 = 5$.

		Persistent TE			Transient TE			Overall TE		
n	t	R Bias	Corr	5th, 50th and 95th percentiles	R Bias	Corr	5th, 50th and 95th percentiles	R Bias	Corr	5th, 50th and 95th percentiles
S7: $\lambda_0 = 5,\ \lambda = 0.2,\ \Lambda = 5$ ($\sigma_{u0} = 0.02\ \sigma_u = 0.04,\ \sigma_{v0} = 0.04,\ \sigma_v = 0.2$)										
100	10	0.074	0.847	(0.245; 0.060; -0.039)	-0.042	0.115	(-0.001; -0.048; -0.068)	0.028	0.707	(0.188; 0.014; -0.085)
100	20	0.049	0.891	(0.163; 0.043; -0.040)	-0.038	0.115	(0.003; -0.043; -0.064)	0.009	0.815	(0.116; 0.003; -0.081)
100	40	0.038	0.916	(0.122; 0.036; -0.036)	-0.034	0.118	(0.007; -0.038; -0.060)	0.003	0.867	(0.086; 0.003; -0.078)
500	10	0.02	0.853	(0.096; 0.018; -0.050)	-0.034	0.114	(0.009; -0.038; -0.060)	-0.014	0.828	(0.058; -0.015; -0.088)
500	20	0.012	0.897	(0.057; 0.018; -0.042)	-0.028	0.117	(0.016; -0.032; -0.055)	-0.016	0.874	(0.034; -0.012; -0.077)
500	40	0.008	0.921	(0.039; 0.013; -0.036)	-0.022	0.118	(0.021; -0.027; -0.049)	-0.015	0.9	(0.024; -0.007; -0.066)
1000	10	0.016	0.854	(0.085; 0.021; -0.050)	-0.03	0.114	(0.013; -0.035; -0.057)	-0.014	0.833	(0.055; -0.013; -0.086)
1000	20	0.009	0.897	(0.049; 0.016; -0.041)	-0.023	0.117	(0.021; -0.029; -0.051)	-0.015	0.877	(0.031; -0.008; -0.072)
1000	40	0.006	0.922	(0.036; 0.013; -0.036)	-0.018	0.119	(0.027; -0.022; -0.045)	-0.012	0.902	(0.021; -0.006; -0.062)
S8: $\lambda_0 = 5,\ \lambda = 1,\ \Lambda = 1$ ($\sigma_{u0} = 0.2\ \sigma_u = 0.2,\ \sigma_{v0} = 0.04,\ \sigma_v = 0.2$)										
100	10	0.084	0.823	(0.284; 0.065; -0.038)	0.036	0.505	(0.243; 0.013; -0.085)	0.122	0.653	(0.400; 0.102; -0.073)
100	20	0.057	0.876	(0.194; 0.049; -0.039)	0.024	0.516	(0.217; 0.003; -0.093)	0.081	0.692	(0.304; 0.064; -0.088)
100	40	0.041	0.908	(0.137; 0.039; -0.037)	0.018	0.521	(0.210; -0.004; -0.098)	0.06	0.721	(0.258; 0.051; -0.093)
500	10	0.025	0.827	(0.120; 0.022; -0.055)	0.016	0.508	(0.204; -0.005; -0.099)	0.04	0.722	(0.220; 0.034; -0.102)
500	20	0.014	0.882	(0.072; 0.019; -0.046)	0.014	0.517	(0.196; -0.008; -0.100)	0.028	0.743	(0.179; 0.015; -0.098)
500	40	0.009	0.913	(0.043; 0.010; -0.037)	0.014	0.521	(0.197; -0.008; -0.099)	0.022	0.753	(0.168; 0.013; -0.097)
1000	10	0.019	0.828	(0.105; 0.017; -0.055)	0.013	0.508	(0.200; -0.007; -0.102)	0.031	0.726	(0.196; 0.023; -0.105)
1000	20	0.011	0.882	(0.062; 0.014; -0.045)	0.013	0.516	(0.195; -0.009; -0.100)	0.023	0.744	(0.170; 0.016; -0.101)
1000	40	0.006	0.914	(0.040; 0.014; -0.038)	0.013	0.522	(0.193; -0.009; -0.100)	0.019	0.754	(0.158; 0.010; -0.097)
S3: $\lambda_0 = 5,\ \lambda = 1,\ \Lambda = 5$ ($\sigma_{u0} = 0.2\ \sigma_u = 0.04,\ \sigma_{v0} = 0.04,\ \sigma_v = 0.04$)										
100	10	0.026	0.939	(0.082; 0.030; -0.031)	0.006	0.505	(0.043; 0.003; -0.018)	0.033	0.923	(0.092; 0.034; -0.032)
100	20	0.025	0.942	(0.079; 0.028; -0.030)	0.003	0.518	(0.039; -0.000; -0.020)	0.029	0.925	(0.083; 0.031; -0.035)

100	40	0.025	0.943	(0.079; 0.029; -0.029)	0.002	0.526	(0.037; -0.000; -0.021)	0.027	0.925	(0.083; 0.030; -0.036)
500	10	0.004	0.943	(0.025; 0.010; -0.030)	0.002	0.506	(0.038; -0.001; -0.021)	0.007	0.932	(0.033; 0.013; -0.033)
500	20	0.005	0.946	(0.026; 0.010; -0.028)	0.001	0.519	(0.036; -0.002; -0.021)	0.006	0.933	(0.027; 0.011; -0.034)
500	40	0.005	0.947	(0.026; 0.009; -0.029)	0.001	0.526	(0.035; -0.003; -0.022)	0.006	0.934	(0.030; 0.013; -0.035)
1000	10	0.003	0.943	(0.020; 0.009; -0.030)	0.002	0.506	(0.037; -0.003; -0.022)	0.005	0.932	(0.028; 0.012; -0.034)
1000	20	0.003	0.946	(0.021; 0.007; -0.029)	0.001	0.519	(0.034; -0.002; -0.022)	0.004	0.934	(0.027; 0.011; -0.034)
1000	40	0.003	0.948	(0.019; 0.007; -0.027)	0.001	0.526	(0.035; -0.003; -0.022)	0.004	0.934	(0.026; 0.009; -0.033)
S4: $\lambda_0 = 5,\ \lambda = 5,\ \Lambda = 1$ ($\sigma_{u0} = 0.2\ \sigma_u = 0.2,\ \sigma_{v0} = 0.04,\ \sigma_v = 0.04$)										
100	10	0.034	0.917	(0.115; 0.034; -0.037)	0.009	0.921	(0.042; 0.015; -0.035)	0.042	0.926	(0.111; 0.045; -0.030)
100	20	0.029	0.931	(0.093; 0.031; -0.034)	0.005	0.936	(0.026; 0.011; -0.031)	0.033	0.93	(0.086; 0.037; -0.029)
100	40	0.026	0.938	(0.083; 0.028; -0.031)	0.003	0.943	(0.018; 0.008; -0.028)	0.029	0.934	(0.078; 0.032; -0.031)
500	10	0.002	0.922	(0.028; 0.008; -0.037)	0.008	0.922	(0.035; 0.013; -0.035)	0.009	0.947	(0.036; 0.011; -0.031)
500	20	0.003	0.935	(0.028; 0.008; -0.032)	0.004	0.936	(0.025; 0.009; -0.030)	0.007	0.948	(0.037; 0.011; -0.031)
500	40	0.004	0.942	(0.026; 0.008; -0.030)	0.003	0.943	(0.021; 0.010; -0.029)	0.007	0.948	(0.037; 0.008; -0.030)
1000	10	0	0.922	(0.025; 0.004; -0.035)	0.008	0.922	(0.034; 0.013; -0.034)	0.007	0.947	(0.034; 0.012; -0.031)
1000	20	0.001	0.936	(0.023; 0.007; -0.032)	0.004	0.936	(0.024; 0.009; -0.031)	0.005	0.948	(0.032; 0.005; -0.031)
1000	40	0.002	0.943	(0.021; 0.007; -0.030)	0.003	0.943	(0.023; 0.009; -0.029)	0.004	0.948	(0.030; 0.006; -0.032)

Table A4 Computational scaling of alternative estimators for the four-component stochastic frontier model

Estimator	Main source of computational burden	Computational scaling
INLA	Likelihood evaluation, sparse latent-Gaussian calculations, and low-dimensional integration	$O\left(K\left(NT + C_{\text{sparse}}(N)\right)\right)$
TK	Repeated latent-variable updates over MCMC iterations	$O(MNT)$
PMCMC	Particle evaluation and resampling within each MCMC iteration	$O(MPNT)$
SML	Repeated simulated-likelihood evaluations during optimization	$O(LRNT)$

Notes: M denotes the number of MCMC iterations, P the number of particles used in PMCMC, R the number of simulation draws used in SML, L the number of likelihood evaluations during numerical optimization, and K the number of hyperparameter by INLA. $C_{\text{sparse}}(N)$ denotes the cost of sparse latent-Gaussian calculations associated with the N-dimensional persistent composite effect. The reported orders summarize the principal sources of computational scaling for each estimator

Appendix B. Model Reparameterization and Numerical Sensitivity Analysis

Appendix B.1. Derivation of the Mean-Centered SF Specification

Let

$$m_\varepsilon = E(\varepsilon_{it}) = -\sigma_u\sqrt{2/\pi}, \qquad m_\xi = E(\xi_i) = -\sigma_{u0}\sqrt{2/\pi}.$$

The centered components are

$$\varepsilon_{it}^c = \varepsilon_{it} - m_\varepsilon, \qquad \xi_i^c = \xi_i - m_\xi,$$

with variances

$$\mathrm{Var}(\varepsilon_{it}^c) = \sigma_v^2 + \left(1 - \frac{2}{\pi}\right)\sigma_u^2,$$

and

$$\mathrm{Var}(\xi_i^c) = \sigma_{v0}^2 + \left(1 - \frac{2}{\pi}\right)\sigma_{u0}^2.$$

Writing

$$\boldsymbol{x}_{it} = (1, \tilde{\boldsymbol{x}}_{it}')', \qquad \boldsymbol{\beta} = (\beta_0, \tilde{\boldsymbol{\beta}}')',$$

the centered intercept satisfies

$$\beta_0^m = \beta_0 + m_\xi + m_\varepsilon,$$

and the original frontier intercept is recovered after estimation as

$$\beta_0 = \beta_0^m - m_\xi - m_\varepsilon.$$

Appendix B.2. Sensitivity to the Gaussian-carrier precision

The diffuse Gaussian-carrier representation introduced in Section 3.3 requires the fixed precision τ_c to be sufficiently small that the Gaussian carrier contributes negligible information about the persistent composite effects. At the same time, choosing an unnecessarily extreme precision provides no inferential advantage and may worsen numerical conditioning. This appendix assesses the choice of τ_c in two complementary ways. We first quantify analytically the additional penalty induced by the Gaussian carrier and then complement this with a Monte Carlo experiment examining the sensitivity of posterior inference to alternative carrier precisions.

B.2.1. Analytical magnitude of the carrier approximation

Under the implemented representation, the persistent composite effect is assigned the Gaussian carrier

$$\xi_i^c \sim N\left(0, \tau_c^{-1}\right),$$

while its skew-normal shape is supplied by the auxiliary likelihood described in Section 3.3. Since τ_c is fixed, the normalizing constant of the Gaussian density is omitted from the posterior kernel. The remaining carrier factor is

$$C_{\tau_c}(\boldsymbol{\xi}^c) = \exp\left[-\frac{\tau_c}{2}\sum_{i=1}^{N}(\xi_i^c)^2\right], \qquad \text{(B.1)}$$

where ξ^c collects the N centered persistent composite effects $\xi^c = (\xi_1^c, \ldots, \xi_N^c)'$. Relative to an exactly flat carrier, the change in the log posterior kernel is

$$\Delta\ell_c = -\frac{\tau_c}{2}\sum_{i=1}^{N}(\xi_i^c)^2. \qquad \text{(B.2)}$$

Equation (B.2) makes clear that the approximation depends jointly on the carrier precision, the cross-sectional dimension, and the scale of the persistent composite effects.

If $\sum_{i=1}^{N}(\xi_i^c)^2 \le R^2$ over a region containing essentially all relevant posterior mass, then

$$|\Delta\ell_c| \le \frac{\tau_c R^2}{2}. \qquad \text{(B.3)}$$

For a prescribed tolerance ϵ, requiring the carrier factor to differ from unity by at most ϵ throughout this region gives the sufficient condition

$$1 - \exp\left(-\frac{\tau_c R^2}{2}\right) \leq \epsilon,$$

or equivalently,

$$\tau_c \leq \frac{-2\log(1-\epsilon)}{R^2}. \qquad \text{(B.4)}$$

Thus, a suitably diffuse precision depends on both the dimension and the scale of the latent component, and there is no universal numerical threshold. This condition bounds the unnormalized carrier factor within the stated region, rather than directly bounding changes in normalized posterior summaries.

For the four-component SF, the persistent composite effect combines independent Gaussian heterogeneity and half-normal inefficiency. Mean centering leaves its variance unchanged:

$$\text{Var}(\xi_i^c) = \sigma_{v0}^2 + \left(1 - \frac{2}{\pi}\right)\sigma_{u0}^2. \qquad \text{(B.5)}$$

Since each centered effect has mean zero,

$$E\left[\sum_{i=1}^{N}(\xi_i^c)^2\right] = N\left[\sigma_{v0}^2 + \left(1 - \frac{2}{\pi}\right)\sigma_{u0}^2\right]. \qquad \text{(B.6)}$$

and the expected log-kernel penalty is

$$E(-\Delta\ell_c) = \frac{\tau_c N}{2}\left[\sigma_{v0}^2 + \left(1 - \frac{2}{\pi}\right)\sigma_{u0}^2\right]. \qquad \text{(B.7)}$$

The expectations in (B.6) and (B.7) are taken under the data-generating distribution at the specified component scales. They provide an ex-ante calibration of the carrier contribution, rather than a posterior-wide bound. Table B.1 evaluates this calibration for the weak, balanced, and strong inefficiency-to-noise configurations defined in Table B.2.

Table B.1. Analytical magnitude of the Gaussian-carrier contribution

Scenario	λ_0	N	Expected squared norm	Expected penalty at 10^{-6}	Calibrated distortion
s13	0.2	100	4.0581	2.03×10^{-6}	0.000203%
s13	0.2	1000	40.5814	2.03×10^{-5}	0.002029%
s1	1	100	0.2181	1.09×10^{-7}	0.000011%
s1	1	1000	2.1814	1.09×10^{-6}	0.000109%
s4	5	100	1.6135	8.07×10^{-7}	0.000081%
s4	5	1000	16.1352	8.07×10^{-6}	0.000807%

Note: Calibrated distortion is $100\{1 - \exp[-E(-\Delta\ell_c)]\}$%. The carrier factor is evaluated at the expected squared norm, rather than averaged over the latent effects.

At $\tau_c = 10^{-6}$, the largest expected penalty is 2.03×10^{-5}, attained in scenario s13 with $N = 1000$. The corresponding calibrated distortion is approximately 0.0020%. The direct carrier contribution is therefore small at these scales. The simulation below evaluates whether this calibration translates into stable inference for the full model.

B.2.2. Sensitivity design and evaluation measures

The experiment uses three configurations from the main Monte Carlo design: s13, s1, and s4, with persistent and transient inefficiency-to-noise ratios of 0.2, 1, and 5, respectively. The ratios are defined by $\lambda_0 = \frac{\sigma_{u0}}{\sigma_{v0}}, \lambda = \frac{\sigma_u}{\sigma_v}$, and their component scales are reported in Table B.2.

Table B.2. Variance configurations used in the sensitivity experiment

Scenario	σ_{v0}	σ_{u0}	λ_0	σ_v	σ_u	λ	Inefficiency relative to noise
s13	0.20	0.04	0.2	0.20	0.04	0.2	Weak
s1	0.04	0.04	1	0.04	0.04	1	Balanced
s4	0.04	0.20	5	0.04	0.20	5	Strong

Note: Weak, balanced, and strong refer to the relative magnitudes of inefficiency and Gaussian noise in both composite components.

Each configuration is evaluated at $(N, T) = (100,10)$ and $(1000,40)$, the smallest and largest panels in the main simulation design. For each of the six resulting designs, 100 datasets are generated, and every dataset is fitted under all six precisions:

$$\tau_c \in \{10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}, 10^{-8}\}. \qquad \text{(B.8)}$$

This paired design isolates sensitivity to τ_c by holding the dataset and all other fitting settings fixed within each replication. The main-analysis value 10^{-6} is compared with the more diffuse reference 10^{-8}. Larger precisions, particularly 10^{-2} and 10^{-3}, assess sensitivity to a more informative carrier.

Frontier parameters

Let $\Psi = \{\beta_0, \beta_1, \beta_2, \sigma_v, \sigma_u, \sigma_{v0}, \sigma_{u0}\}$ denote the structural parameter vector, and let the estimated value of each ψ be its posterior mean in replication r. Relative to 10^{-8}, we calculate the difference

$$D_{\psi,r}(\tau_c) = \widehat{\psi}_r(\tau_c) - \widehat{\psi}_r(10^{-8}), \qquad \text{(B.9)}$$

the absolute relative difference

$$R_{\psi,r}(\tau_c) = \frac{|\widehat{\psi}_r(\tau_c) - \widehat{\psi}_r(10^{-8})|}{|\widehat{\psi}_r(10^{-8})|}. \qquad \text{(B.10)}$$

and the absolute posterior-standardized difference

$$D^S_{\psi,r}(\tau_c) = \frac{|\widehat{\psi}_r(\tau_c) - \widehat{\psi}_r(10^{-8})|}{SD_r[\psi \mid D_r, \tau_c = 10^{-8}]}. \qquad \text{(B.11)}$$

All measures are averaged over replications. Standardized differences are interpreted alongside absolute and relative differences because a small numerical change can represent a larger fraction of posterior uncertainty in highly concentrated posteriors.

Latent effects and efficiency

We compare posterior means of the stochastic component and their TE predictions. For the centered persistent composite effect, the within-replication mean absolute difference is

$$A_{\xi,r}(\tau_c) = \frac{1}{N}\sum_{i=1}^{N} |\hat{\xi}^c_{i,r}(\tau_c) - \hat{\xi}^c_{i,r}(10^{-8})|. \qquad \text{(B.12)}$$

Analogous measures are calculated for PTE and RTE. For PTE,

$$A_{P,r}(\tau_c) = \frac{1}{N}\sum_{i=1}^{N} |\widehat{TE}^P_{i,r}(\tau_c) - \widehat{TE}^P_{i,r}(10^{-8})|, \qquad \text{(B.13)}$$

with corresponding measures for RTE and OTE. We also assess recovery against the simulated truth using correlations and RMSEs for ξ^c_i, u_{0i}, v_{0i}, u_{it}, and v_{it}, and RMSEs for all three TE measures.

Skewness and numerical performance

Because the skewness parameters are estimated without sign restrictions, we compare $P(\gamma_\varepsilon < 0|\mathcal{D}_r)$ and $P(\gamma_\xi < 0|\mathcal{D}_r)$, together with the posterior means of γ_ε and γ_ξ. These diagnostics assess sensitivity of the estimated skewness orientation. Numerical performance is evaluated by the INLA fit-success rate and core computation time.

B.2.3. Sensitivity results

Frontier parameters

Comparing 10^{-6} with 10^{-8}, the average absolute relative difference across the 42 parameter-design combinations is 0.58%, and the average absolute standardized difference is 0.0476 posterior standard deviations. No mean standardized difference exceeds 0.22. Table B.3 summarizes the variation across designs.

Table B.3. Structural-parameter sensitivity of 10^{-6} relative to 10^{-8}

Scenario	Panel (N,T)	Mean S	Largest mean S	Parameter at largest S	Largest mean R
s13	(100, 10)	0.029	0.046	σ_{v0}	1.95%
s13	(1000, 40)	0.104	0.217	σ_{u0}	5.11%
s1	(100, 10)	0.027	0.039	σ_{u0}	1.41%
s1	(1000, 40)	0.037	0.060	σ_{v0}	1.12%
s4	(100, 10)	0.028	0.045	σ_{v0}	0.74%

Scenario	Panel (N,T)	Mean S	Largest mean S	Parameter at largest S	Largest mean R
s4	(1000, 40)	0.060	0.104	σ_{v0}	0.62%

Note: S and R are defined in Eqs. (B.11) and (B.10). Mean S averages over the seven parameters. The largest-value columns report maxima of parameter-specific replication means; the parameter column identifies the maximum of S, which need not coincide with the maximum of R.

The largest parameter sensitivity concerns σ_{u0} in the large s13 panel. The mean absolute difference is 0.00528, the mean relative difference is 5.11%, and the mean standardized difference is 0.217, with a 95th percentile of 0.841. This configuration has $\lambda_0 = 0.2$ and weak persistent skewness, as shown below. The concentration of sensitivity in this component is consistent with difficulty separating persistent inefficiency from time-invariant heterogeneity. The regression slopes β_1 and β_2 show much smaller changes relative to their posterior uncertainty.

Latent effects and efficiency

Averaged over the 6 designs, the mean absolute differences are 5.2×10^{-5} for ξ_i^c, 9.35×10^{-4} for u_{0i}, and 3.28×10^{-4} for u_{it}. The corresponding differences in PTE, RTE and OTE are 8.41×10^{-4}, 3.02×10^{-4}, and 8.93×10^{-4}, respectively (Table B.4).

Table B.4. Mean absolute differences in TE predictions

Scenario	Panel (N,T)	PTE	RTE	OTE
s13	(100, 10)	0.000891	0.000622	0.001016
s13	(1000, 40)	0.002862	0.000907	0.003092
s1	(100, 10)	0.000262	0.000150	0.000314
s1	(1000, 40)	0.000257	0.000022	0.000248
s4	(100, 10)	0.000553	0.000087	0.000493
s4	(1000, 40)	0.000221	0.000022	0.000196

The largest change in overall efficiency is 0.00309 in the large s13 panel, or approximately 0.31 percentage points on the unit interval. Recovery against the simulated truth is also stable: the largest change across the reported recovery statistics is 7.86×10^{-4}, for the RMSE of v_{0i}. The maximum RMSE changes for persistent and transient inefficiency are 5.8×10^{-5} and 1.50×10^{-4}; those for persistent, transient, and overall efficiency are 5.5×10^{-5}, 1.30×10^{-4}, and 9.6×10^{-5}. Thus, the larger scale sensitivity in s13 has little effect on the reported recovery measures.

Skewness orientation

The largest mean absolute changes in the posterior probabilities of negative transient and persistent skewness are 0.0192 and 0.0464. The largest mean changes in the corresponding skewness coefficients are 0.00139 and 0.00498. Table B.5 reports the diagnostics.

Table B.5. Posterior skewness diagnostics at 10^{-6}

Scenario	Panel (N,T)	$Pr(\gamma_\varepsilon < 0 \mid y)$	$Pr(\gamma_\xi < 0 \mid y)$	Mean γ_ε	Mean γ_ξ
s13	(100, 10)	0.550	0.504	-0.0093	-0.0012
s13	(1000, 40)	0.571	0.512	-0.0029	-0.0014
s1	(100, 10)	0.884	0.624	-0.1035	-0.0523
s1	(1000, 40)	1.000	0.943	-0.1303	-0.1132
s4	(100, 10)	1.000	0.973	-0.8070	-0.5549
s4	(1000, 40)	1.000	1.000	-0.8464	-0.8322

Note: Entries are averages over replications. γ_ε and γ_ξ denote the transient and persistent standardized skewness coefficients.

The posterior sign probabilities vary mainly with signal strength. They are close to one-half in s13, increase with panel size in s1, and approach unity in s4. This pattern supports interpreting the greater sensitivity in s13 in the context of weak separation of the persistent components; it does not by itself establish the cause of the observed parameter differences.

Full precision sequence and numerical performance

Table B.6 reports sensitivity across all 6 precisions. For each replication, the maximum standardized difference is taken over the seven structural parameters; these maxima are then averaged over replications and designs.

Table B.6. Sensitivity and numerical performance across carrier precisions

τ_c	Mean max S	Persistent TE MAD	Transient TE MAD	Overall TE MAD	Running time (seconds)	Success rate
10^{-8}	0.000	0.000000	0.000000	0.000000	14.376	1.000
10^{-6}	0.125	0.000841	0.000302	0.000893	14.384	1.000
10^{-5}	0.123	0.000709	0.000293	0.000766	14.387	1.000
10^{-4}	0.122	0.000805	0.000295	0.000845	14.380	1.000
10^{-3}	0.136	0.000844	0.000320	0.000906	14.358	1.000
10^{-2}	0.177	0.000996	0.000359	0.001017	14.364	1.000

Note: Differences are measured relative to 10^{-8}. MAD denotes the mean absolute difference in posterior-mean efficiency predictions. Entries are averaged over the 6 designs; S is defined in (B.11).

The mean maximum standardized difference is 0.177 at 10^{-2}, compared with 0.125 at 10^{-6}; it reaches 0.447 at 10^{-2} in the most sensitive design. Across 10^{-3}–10^{-6}, the aggregate measures show no systematic improvement as the carrier becomes more diffuse. This suggests practical stability, although the experiment does not distinguish carrier perturbation from numerical approximation and posterior-sampling variation in the remaining differences.

All $3{,}600$ INLA fits complete successfully, and mean running computation time ranges from approximately 14.36 to 14.39 seconds across precisions. Reducing 10^{-6} to 10^{-8} yields no improvement in these computation measures.

In summary, all results above support $\tau_c = 10^{-6}$ for the dimensions and scales examined. The direct carrier contribution is small, posterior inference and efficiency predictions are stable, and greater scale sensitivity in s13 has little effect on recovery. We retain this value in the main analysis.

Appendix C. Technical derivations for posterior recovery and efficiency prediction

C.1. From standardized skewness to the skew-normal scale

For a centered skew-normal distribution with standardized skewness γ (Azzalini & Valle, 1996), the relationship between γ and the bounded shape parameter $\delta \in (-1,1)$ is

$$\gamma = \frac{4-\pi}{2}\frac{\left(\delta\sqrt{2/\pi}\right)^3}{\left(1-2\delta^2/\pi\right)^{3/2}}. \qquad \text{(B.1)}$$

Define

$$A = \frac{4-\pi}{2}, \qquad q = \left(\frac{|\gamma|}{A}\right)^{2/3},$$

inverting Eq. B.1 gives

$$\delta = \text{sign}(\gamma)\sqrt{\frac{\pi}{2}\frac{q}{1+q}}. \qquad \text{(B.2)}$$

For a mean-centered skew-normal variable, its variance s^2 and the scale ω of the corresponding uncentered skew-normal distribution satisfy

$$s^2 = \omega^2\left(1-\frac{2\delta^2}{\pi}\right). \qquad \text{(B.3)}$$

Hence,

$$\omega^2 = \frac{s^2}{1-2\delta^2/\pi}. \qquad \text{(B.4)}$$

Combining Eqs. B.2-B.4 gives the transformation used in Section 3.5 to recover the original normal-half normal scale parameters from the precision and standardized skewness of each centered skew-normal likelihood.

C.2. Conditional distribution of inefficiency given the composite variance component

Consider

$$z = g - h,$$

where

$$g \sim N(0,\sigma_g^2), \qquad h \sim N^+(0,\sigma_h^2),$$

and g and h are independent. Up to a normalizing constant, their joint density is

$$p(g,h) \propto \exp\left(-\frac{g^2}{2\sigma_g^2}-\frac{h^2}{2\sigma_h^2}\right)\mathbf{1}(h \geq 0). \qquad \text{(B.5)}$$

Since $g = z + h$, conditioning on z gives

$$p(h|z) \propto \exp\left[-\frac{(z+h)^2}{2\sigma_g^2}-\frac{h^2}{2\sigma_h^2}\right]\mathbf{1}(h \geq 0). \qquad \text{(B.6)}$$

Completing the square in h yields

$$p(h|z) \propto \exp\left[-\frac{(h-\mu_h^*)^2}{2\sigma_h^{*2}}\right]\mathbf{1}(h \geq 0), \qquad \text{(B.7)}$$

where

$$\mu_h^* = -\frac{\sigma_h^2}{\sigma_g^2+\sigma_h^2}z, \qquad \sigma_h^{*2} = \frac{\sigma_g^2\sigma_h^2}{\sigma_g^2+\sigma_h^2}. \qquad \text{(B.8)}$$

Therefore,

$$h|z \sim N^+\left(\mu_h^*, \sigma_h^{*2}\right). \qquad \text{(B.9)}$$

Its conditional mean is

$$E(h|z) = \mu_h^* + \sigma_h^*\frac{\phi(\mu_h^*/\sigma_h^*)}{\Phi(\mu_h^*/\sigma_h^*)}. \qquad \text{(B.10)}$$

Eq. B.10 is the normal-half normal conditional inefficiency predictor of Jondrow et al. (1982).

For the persistent component,

$$(z, g, h) = (\xi_i, v_{0i}, u_{0i}),$$

while for the transient component,

$$(z, g, h) = (\varepsilon_{it}, v_{it}, u_{it}).$$

Thus, Eq. B.10 applies directly to both dimensions of inefficiency in the four-component model.

C.3. Conditional technical efficiency

Technical efficiency is defined as $\exp(-h)$ (see section 3.1). For the truncated-normal conditional distribution in Eq. B.9,

$$E[\exp(-h)|z] = \exp\left(-\mu_h^* + \frac{\sigma_h^{*2}}{2}\right)\frac{\Phi\left[(\mu_h^* - \sigma_h^{*2})/\sigma_h^*\right]}{\Phi(\mu_h^*/\sigma_h^*)}. \qquad \text{(B.11)}$$

Eq. B.11 follows from the moment-generating function of a normal distribution truncated below at zero and gives the conditional TE predictor. In the stochastic frontier literature, prediction of technical efficiency conditional on the composite disturbance is associated with Battese & Coelli (1988).

For the persistent component, equation (B.11) gives

$$PTE_i = E[\exp(-u_{0i})|\xi_i], \qquad \text{(B.12)}$$

and for the transient component,

$$RTE_{it} = E[\exp(-u_{it})|\varepsilon_{it}], \qquad \text{(B.13)}$$

In posterior computation, Eqs B.10-B.13 are evaluated at each joint posterior draw of the structural parameter coefficients, composite components, and scale parameters and subsequently averaged as described in Section 3.5.

Reference

Azzalini, A., & Valle, A. D. (1996). The multivariate skew-normal distribution. *Biometrika*, *83*(4), 715–726. https://doi.org/10.1093/biomet/83.4.715

Battese, G. E., & Coelli, T. J. (1988). Prediction of firm-level technical efficiencies with a generalized frontier production function and panel data. *Journal of Econometrics*, *38*(3), 387–399. https://doi.org/https://doi.org/10.1016/0304-4076(88)90053-X

Jondrow, J., Knox Lovell, C. A., Materov, I. S., & Schmidt, P. (1982). On the estimation of technical inefficiency in the stochastic frontier production function model. *Journal of Econometrics*, *19*(2), 233–238. https://doi.org/https://doi.org/10.1016/0304-4076(82)90004-5

Appendix D Comparison of INLA and FIML estimates across two empirical cases

To assess the robustness of the proposed INLA estimator, this section compares its posterior estimates with full-information maximum likelihood (FIML) estimates under the same four-component specification for the two empirical applications presented in Section 5.

Table D1 compares INLA posterior means and standard deviations with FIML estimates and standard errors for the same four-component specification. The input elasticities agree closely: differences are at most 0.0014 for farm dataset and 0.0001 for firm dataset. The intercept estimates are also similar, at 11.7447 and 11.7609. This agreement supports the robustness of the estimated frontier coefficients.

Component scales show larger differences, particularly for persistent inefficiency. For the farm case, INLA and FIML estimate σ_{u0} at 0.1315 and 0.1467, respectively, and σ_{v_0} at 0.0837 and 0.0742, preserving the same relative ordering. For the firm case, the persistent inefficiency estimates are 0.1663 and 0.2411, although the FIML standard error is substantial at 0.2558. The remaining component scales agree more closely.

Overall, the FIML results support the robustness of the INLA estimates, particularly for the frontier coefficients and the broad pattern of component scales. The larger differences in persistent inefficiency indicate that its precise magnitude is less stable across estimation methods, especially in the firm case, where persistent inefficiency is relatively small compared with time-invariant heterogeneity.

Table D1 Comparison of INLA posterior estimates and FIML estimates for both frontier and scale parameters

Parameter	FIML estimates (SE)	INLA posterior mean (SD)	$INLA - FIML$
Panel A			
Intercept	11.7609 (0.0167)	11.7447 (0.0117)	-0.0162
β_{cows}	0.6407 (0.0205)	0.6421 (0.0205)	0.0014
β_{land}	0.0357 (0.0133)	0.0348 (0.0132)	-0.0009
β_{labor}	0.0346 (0.017)	0.0351 (0.0171)	0.0005
β_{feed}	0.399 (0.0108)	0.3991 (0.0108)	0.0001
σ_{v0}	0.0742 (0.0112)	0.0837 (0.006)	0.0095
σ_{u0}	0.1467 (0.0191)	0.1315 (0.0113)	-0.0152
σ_v	0.0633 (0.0038)	0.0663 (0.002)	0.0030
σ_u	0.0842 (0.0083)	0.0781 (0.0047)	-0.0061
Panel B			
Intercept	-3.6555 (0.2064)	-3.7196 (0.0702)	-0.0641
β_{labor}	0.3549 (0.0026)	0.3549 (0.0026)	0.0000
$\beta_{capital}$	0.3035 (0.002)	0.3035 (0.0019)	0.0000
$\beta_{materials}$	0.3026 (0.0022)	0.3027 (0.0022)	0.0001
σ_{v0}	0.412 (0.0556)	0.4222 (0.0129)	0.0102
σ_{u0}	0.2411 (0.2558)	0.1663 (0.0795)	-0.0748
σ_v	0.1919 (0.0066)	0.1952 (0.0032)	0.0033

σ_u	0.2978 (0.0129)	0.2929 (0.0063)	-0.0049

Notes: Parentheses report standard errors for FIML and posterior standard deviations for INLA. The final column reports the difference between the INLA posterior mean and the FIML point estimate. The comparison assesses sensitivity to the estimation approach rather than serving as a test of latent-efficiency accuracy.

Appendix E.1 Estimating a Four-Component Stochastic Frontier Models with INLA

Empirical Illustration 1 — Spanish Dairy Farms

2026-10-16

```
install.packages(c("rmarkdown", "knitr", "tinytex", "sfaR", "ggplot2"),
                 repos = "https://cloud.r-project.org")
install.packages("INLA", repos = c(
  CRAN = "https://cloud.r-project.org",
  INLA = "https://inla.r-inla-download.org/R/testing"
), dependencies = TRUE)
# Only if a working LaTeX distribution is not already installed:
tinytex::install_tinytex()

# Alternative to the Knit button, run in the R console:
rmarkdown::render("Dairyspain_INLA_report_v10.Rmd")
```

```
rmd_path <- knitr::current_input(dir = TRUE)
document_dir <- if (length(rmd_path) == 1L && nzchar(rmd_path)) {
  dirname(normalizePath(rmd_path, mustWork = TRUE))
} else {
  getwd()
}
helper_path <- file.path(document_dir, "Dairyspain_production_helpers.R")
if (!file.exists(helper_path)) {
  stop("Missing external helper: ", helper_path,
       ". Put Dairyspain_production_helpers.R beside this Rmd.")
}
source(helper_path, local = TRUE)
if (!exists("prepare_production_draws", mode = "function", inherits = FALSE)) {
  stop("The helper file did not define prepare_production_draws().")
}
```

1 Model and inferential scope

1.1 Analysis settings

The original prior settings, 12,000 requested draws, and random seeds are retained. Fixed effects have Gaussian priors with precision 10^{-4}. The carrier precision is 10^{-6}. Negative starting skewness values are initial values, not sign constraints.

```
RUN_PRODUCTION_TE <- TRUE
POSTERIOR_DRAWS <- 12000L
BASE_SEED <- 20261004L
PC_COMPONENT_SD_UPPER <- 1.0
PC_COMPONENT_SD_PROB <- 0.01
```

```r
PC_SKEW_PARAM <- 10
CARRIER_PRECISION <- 1e-6
FIXED_EFFECT_PRECISION <- 1e-4
INITIAL_DELTA_EPSILON <- -0.80
INITIAL_DELTA_XI <- -0.80
INLA_NUM_THREADS <- "1:1"
INLA_VERBOSE <- FALSE
MAKE_PLOTS <- TRUE

if (length(POSTERIOR_DRAWS) != 1L || !is.finite(POSTERIOR_DRAWS) ||
    POSTERIOR_DRAWS < 2L || POSTERIOR_DRAWS != floor(POSTERIOR_DRAWS)) {
  stop("POSTERIOR_DRAWS must be an integer of at least two.")
}
if (any(!is.finite(c(PC_COMPONENT_SD_UPPER, PC_COMPONENT_SD_PROB,
                     PC_SKEW_PARAM, CARRIER_PRECISION, FIXED_EFFECT_PRECISION))) ||
    PC_COMPONENT_SD_UPPER <= 0 || PC_COMPONENT_SD_PROB <= 0 ||
    PC_COMPONENT_SD_PROB >= 1 || PC_SKEW_PARAM <= 0 ||
    CARRIER_PRECISION <= 0 || FIXED_EFFECT_PRECISION <= 0) {
  stop("Invalid precision or PC-prior settings.")
}

# Dairyspain FIML results transcribed from parameter.docx:
# Dataset 1, robustness-check comparison table (page 3).
# These are reported estimates, not a new FIML fit in this document.
# Use sigma_v0 for the persistent Gaussian heterogeneity scale.
FIML_INPUT <- data.frame(
  parameter = c(
    "beta0", "beta_cows", "beta_land", "beta_labor", "beta_feed",
    "sigma_v0", "sigma_u0", "sigma_v", "sigma_u"
  ),
  estimate = c(
    11.7609, 0.6407, 0.0357, 0.0346, 0.3990,
    0.0742, 0.1467, 0.0633, 0.0842
  ),
  standard_error = c(
    0.0167, 0.0205, 0.0133, 0.0170, 0.0108,
    0.0112, 0.0191, 0.0038, 0.0083
  ),
  stringsAsFactors = FALSE
)
stopifnot(nrow(FIML_INPUT) == 9L,
          all(is.finite(FIML_INPUT$estimate)),
          all(is.finite(FIML_INPUT$standard_error)))
```

```r
C_HALF_NORMAL <- sqrt(2 / pi)
INLA_SKEW_MAX <- 0.988

center_only <- function(x) {
  as.numeric(x - mean(x))
}

summarize_draws <- function(x) {
  x <- x[is.finite(x)]
  if (length(x) == 0L) {
    return(c(mean = NA_real_, sd = NA_real_, q0.025 = NA_real_,
             q0.500 = NA_real_, q0.975 = NA_real_))
```

```
  }

  c(
    mean = mean(x),
    sd = sd(x),
    q0.025 = unname(quantile(x, 0.025)),
    q0.500 = unname(quantile(x, 0.500)),
    q0.975 = unname(quantile(x, 0.975))
  )
}
```

1.2 Skew-normal transformations

```
gamma_from_delta <- function(delta) {

  numerator <-
    ((4 - pi) / 2) *
    (delta * C_HALF_NORMAL)^3

  denominator <-
    (1 - 2 * delta^2 / pi)^(3 / 2)

  numerator / denominator
}

delta_from_gamma <- function(gamma) {

  answer <- rep(NA_real_, length(gamma))

  valid <- is.finite(gamma)

  gamma_valid <- gamma[valid]

  A <- (4 - pi) / 2

  q <-
    (abs(gamma_valid) / A)^(2 / 3)

  delta_squared <-
    (pi / 2) * q / (1 + q)

  delta_squared <-
    pmin(delta_squared, 1 - 1e-12)

  answer[valid] <-
    sign(gamma_valid) *
    sqrt(delta_squared)

  answer[
    valid & abs(gamma) < 1e-14
  ] <- 0

  answer
}
```

```r
skew_to_internal_theta <- function(
    gamma,
    skew_max = INLA_SKEW_MAX
) {

  gamma <- max(
    min(gamma, 0.98 * skew_max),
    -0.98 * skew_max
  )

  log(
    (1 + gamma / skew_max) /
      (1 - gamma / skew_max)
  )
}

map_sn_draws_to_components <- function(
    precision,
    gamma
) {

  delta <- delta_from_gamma(gamma)

  centered_variance <- 1 / precision

  total_variance <-
    centered_variance /
    (1 - 2 * delta^2 / pi)

  sigma_total <- sqrt(total_variance)

  sigma_gaussian <-
    sigma_total *
    sqrt(1 - delta^2)

  sigma_halfnormal <-
    sigma_total *
    abs(delta)

  mean_shift <-
    sigma_total *
    delta *
    C_HALF_NORMAL

  data.frame(
    precision = precision,
    gamma = gamma,
    delta = delta,
    sigma_total = sigma_total,
    sigma_gaussian = sigma_gaussian,
    sigma_halfnormal = sigma_halfnormal,
    mean_shift = mean_shift,
    lambda =
      sigma_halfnormal /
      pmax(
        sigma_gaussian,
```

```
        .Machine$double.eps
      )
  )
}
```

1.3 Conditional inefficiency and efficiency

```
conditional_halfnormal_parameters <- function(
    composed_raw,
    sigma_gaussian,
    sigma_halfnormal
) {

  vg <- sigma_gaussian^2
  vh <- sigma_halfnormal^2

  denominator <- vg + vh

  variance_cond <-
    vg * vh / denominator

  sd_cond <-
    sqrt(
      pmax(variance_cond, 0)
    )

  mean_cond <-
    -vh / denominator *
    composed_raw

  list(
    mean = mean_cond,
    sd = sd_cond
  )
}

# Survival-probability inversion avoids subtracting two probabilities near 1.
rtruncnorm_lower0 <- function(n, mean, sd) {
  mean <- rep_len(mean, n)
  sd <- rep_len(sd, n)
  if (any(!is.finite(mean) | !is.finite(sd) | sd < 0)) {
    stop("Invalid conditional truncated-normal parameters.")
  }
  ans <- pmax(mean, 0)
  positive <- sd > 0
  if (any(positive)) {
    a <- -mean[positive] / sd[positive]
    log_survival <- pnorm(a, lower.tail = FALSE, log.p = TRUE)
    log_p <- log_survival + log(pmax(runif(sum(positive)), .Machine$double.xmin))
    z <- qnorm(log_p, lower.tail = FALSE, log.p = TRUE)
    ans[positive] <- pmax(mean[positive] + sd[positive] * z, 0)
  }
  if (any(!is.finite(ans))) stop("Nonfinite inefficiency draw.")
  ans
```

```r
}

conditional_halfnormal_te <- function(
    composed_raw,
    sigma_gaussian,
    sigma_halfnormal
) {
  if (is.finite(sigma_halfnormal) && sigma_halfnormal == 0) return(rep(1,
   ↪  length(composed_raw)))
  stopifnot(
    length(sigma_gaussian) == 1L,
    length(sigma_halfnormal) == 1L,
    is.finite(sigma_gaussian),
    is.finite(sigma_halfnormal),
    sigma_gaussian > 0,
    sigma_halfnormal > 0
  )

  variance_gaussian <- sigma_gaussian^2
  variance_halfnormal <- sigma_halfnormal^2

  conditional_variance <-
    variance_gaussian * variance_halfnormal /
    (variance_gaussian + variance_halfnormal)

  conditional_sd <- sqrt(conditional_variance)

  conditional_mean <-
    -variance_halfnormal /
    (variance_gaussian + variance_halfnormal) * composed_raw

  log_te <-
    -conditional_mean +
    0.5 * conditional_variance +
    pnorm(
      (conditional_mean - conditional_variance) / conditional_sd,
      log.p = TRUE
    ) -
    pnorm(
      conditional_mean / conditional_sd,
      log.p = TRUE
    )

  answer <- exp(log_te)
  answer[!is.finite(answer)] <- NA_real_

  # Mathematical bounds; the clamp only removes tiny numerical error.
  pmin(1, pmax(0, answer))
}
```

2 Data preparation

2.1 Load, validate, and index the panel

```
data_env <- new.env()
utils::data("dairyspain", package = "sfaR", envir = data_env)
if (!exists("dairyspain", envir = data_env, inherits = FALSE)) stop("Cannot load dairyspain.")
raw <- as.data.frame(data_env$dairyspain)
required <- c("FARM", "YEAR", "YIT", "X1", "X2", "X3", "X4")
if (!all(required %in% names(raw))) stop("Unexpected dairyspain columns: ",
 ↪  paste(setdiff(required, names(raw)), collapse = ", "))
if (anyNA(raw[required])) stop("Missing model data: no rows were silently removed.")
log_columns <- c("YIT", "X1", "X2", "X3", "X4")
if (!all(vapply(raw[log_columns], is.numeric, logical(1))) ||
    any(!is.finite(as.matrix(raw[log_columns])))) stop("Model log columns must be finite
     ↪  numeric values.")
if (any(!nzchar(trimws(as.character(raw$FARM))))) stop("Empty farm identifiers.")
year_numeric <- suppressWarnings(as.numeric(as.character(raw$YEAR)))
if (any(!is.finite(year_numeric)) || any(year_numeric != floor(year_numeric))) stop("Invalid
 ↪  year values.")
# The source stores years as 93:98; label them as 1993:1998.
if (all(year_numeric %in% 93:98)) year_numeric <- year_numeric + 1900
# Record original source rows before sorting.
dat <- data.frame(original_row = seq_len(nrow(raw)), farm_id = as.character(raw$FARM),
  year = as.integer(year_numeric), log_output = raw$YIT,
  log_cows = raw$X1, log_land = raw$X2, log_labor = raw$X3, log_feed = raw$X4)
if (anyDuplicated(dat[c("farm_id", "year")])) stop("Duplicate farm-year records.")
dat <- dat[order(dat$farm_id, dat$year), , drop = FALSE]
rownames(dat) <- NULL
farm_ids <- sort(unique(dat$farm_id))
dat$id <- match(dat$farm_id, farm_ids)
N <- length(farm_ids); NT <- nrow(dat); TT <- length(unique(dat$year))
if (TT < 2L || !all(table(dat$farm_id) == TT)) stop("GTRE/FIML requires a balanced panel with
 ↪  at least two periods.")
regressor_names <- c("log_cows", "log_land", "log_labor", "log_feed")
beta_names <- c("beta_cows", "beta_land", "beta_labor", "beta_feed")
frontier_formula <- log_output ~ log_cows + log_land + log_labor + log_feed
if (qr(model.matrix(frontier_formula, dat))$rank != 5L) stop("Rank-deficient regression
 ↪  design.")
panel_summary <- data.frame(farms = N, years = TT, observations = NT,
  first_year = min(dat$year), last_year = max(dat$year), balanced = TRUE)
print(panel_summary)
```

Table 1: Estimation sample

Farms	Years	Observations	First	Last
247	6	1482	1993	1998

2.2 Center the inputs

```
# Center ALL regressors for fitting; recover intercept on original log/level scale.
input_names <- regressor_names
```

```r
input_means <- colMeans(dat[input_names])
centered_names <- paste0(regressor_names, "_c")
for (nm in input_names) dat[[paste0(nm, "_c")]] <- dat[[nm]] - input_means[nm]
dat$y <- dat$log_output
X <- cbind(1, as.matrix(dat[centered_names]))
```

3 Construct and fit the INLA model

3.1 Real and artificial likelihood rows

```r
# Real likelihood: epsilon_c = v-u-E[v-u].
# Pseudo likelihood: xi_c = v0-u0-E[v0-u0].
Y <- matrix(NA_real_, NT + N, 2L)
Y[seq_len(NT), 1L] <- dat$y
Y[NT + seq_len(N), 2L] <- 0
inla_data <- list(Y = Y,
  int_obs = c(rep(1, NT), rep(0, N)),
  id_xi = c(dat$id, seq_len(N)), w_xi = c(rep(1, NT), rep(-1, N)))
for (nm in centered_names) inla_data[[nm]] <- c(dat[[nm]], rep(0, N))
```

3.2 Native priors

```r
make_native_skew_hyper <- function(initial_delta) {
  if (!is.finite(initial_delta) || initial_delta >= 0 || initial_delta <= -1) {
    stop("Invalid initial delta.")
  }
  list(initial = skew_to_internal_theta(gamma_from_delta(initial_delta)),
       prior = "pc.sn", param = PC_SKEW_PARAM, fixed = FALSE)
}
epsilon_family <- list(hyper = list(
  prec = list(prior = "pc.prec", param = c(PC_COMPONENT_SD_UPPER, PC_COMPONENT_SD_PROB), fixed
  ↪  = FALSE),
  skew = make_native_skew_hyper(INITIAL_DELTA_EPSILON)))
xi_family <- list(hyper = list(
  prec = list(prior = "pc.prec", param = c(PC_COMPONENT_SD_UPPER, PC_COMPONENT_SD_PROB), fixed
  ↪  = FALSE),
  skew = make_native_skew_hyper(INITIAL_DELTA_XI)))
```

3.3 Gaussian carrier and model formula

```r
formula_inla <- as.formula(paste(
  "Y ~ 0 + int_obs +", paste(centered_names, collapse = " + "),
  "+ f(id_xi, w_xi, model = 'iid', constr = FALSE,",
  "hyper = list(prec = list(initial = log(CARRIER_PRECISION), fixed = TRUE)))"))
fit_start <- proc.time()[["elapsed"]]
```

3.4 Fit the model

```
fit <- INLA::inla(
    formula_inla, data = inla_data, family = c("sn", "sn"),
    control.family = list(epsilon_family, xi_family),
    control.fixed = list(mean = 0, prec = FIXED_EFFECT_PRECISION),
    control.predictor = list(compute = TRUE),
    # DIC/WAIC/CPO include pseudo observations, so are not used for FIML comparison.
    control.compute = list(mlik = TRUE, dic = TRUE, waic = TRUE, cpo = TRUE, config = TRUE),
    control.inla = list(strategy = "laplace", int.strategy = "ccd"),
    num.threads = INLA_NUM_THREADS, verbose = INLA_VERBOSE
  )
fit_seconds <- proc.time()[["elapsed"]] - fit_start
```

Table 2: INLA skew-normal hyperparameters

Parameter	mean	sd	0.025quant	0.5quant	0.975quant
precision for skew-normal observations	151.1264	6.1861	139.2765	151.0128	163.6289
Skewness for skew-normal observations	-0.1982	0.0685	-0.3320	-0.1985	-0.0627
precision for skew-normal observations[2]	74.9594	7.5635	60.9440	74.6592	90.6884
Skewness for skew-normal observations[2]	-0.3322	0.1499	-0.6074	-0.3383	-0.0230

4 Joint posterior analysis

4.1 Identify hyperparameters and sample jointly

```
hyper_names <- rownames(fit$summary.hyperpar)

precision_indices <- sort(grep("precision.*skew-normal|skew-normal.*precision",
                               hyper_names, ignore.case = TRUE))
skewness_indices <- sort(grep("skewness.*skew-normal|skew-normal.*skewness",
                              hyper_names, ignore.case = TRUE))
if (length(precision_indices) != 2L || length(skewness_indices) != 2L ||
    length(hyper_names) != 4L) {
  stop("Cannot identify the two native SN pairs; inspect the hyperparameter summary.")
}
# INLA reports the first likelihood before the second likelihood.
hyper_mapping <- data.frame(
  channel = c("epsilon", "xi"),
  precision_name = hyper_names[precision_indices],
  skewness_name = hyper_names[skewness_indices]
)
print(hyper_mapping)

set.seed(BASE_SEED)
post_start <- proc.time()[["elapsed"]]
samples <- INLA::inla.posterior.sample(
  n = POSTERIOR_DRAWS, result = fit, intern = FALSE,
  use.improved.mean = TRUE, seed = BASE_SEED
)
```

```
#> Warning in inla.posterior.sample(n, rfake, intern = intern, use.improved.mean =
#> use.improved.mean, : Since 'seed!=0', parallel model is disabled and serial model is
#> selected, num.threads='1:1'
```

```
# Use the same evaluation mechanism as the supplied simulation script.
evaluated <- t(INLA::inla.posterior.sample.eval(function() {
  c(int_obs, log_cows_c, log_land_c, log_labor_c, log_feed_c, id_xi, theta)
}, samples = samples))
fixed_names <- c("int_obs", centered_names)
P <- length(fixed_names)
if (ncol(evaluated) != P + N + length(hyper_names)) {
  stop("Unexpected joint posterior sample dimension.")
}
colnames(evaluated) <- c(fixed_names, paste0("xi_c_", seq_len(N)), hyper_names)
```

Table 3: Likelihood-channel ordering (first: transient; second: persistent)

Channel	Precision	Skewness
epsilon	SN precision 1	SN skewness 1
xi	SN precision 2	SN skewness 2

4.2 Numerical validity and skewness probabilities

```
rm(samples)
xi_names <- paste0("xi_c_", seq_len(N))
theta_start <- P + N
epsilon_precision <- evaluated[, theta_start + precision_indices[1L]]
xi_precision <- evaluated[, theta_start + precision_indices[2L]]
epsilon_gamma <- evaluated[, theta_start + skewness_indices[1L]]
xi_gamma <- evaluated[, theta_start + skewness_indices[2L]]
basic_valid <- apply(evaluated, 1L, function(z) all(is.finite(z))) &
  epsilon_precision > 0 & xi_precision > 0 &
  abs(epsilon_gamma) < INLA_SKEW_MAX & abs(xi_gamma) < INLA_SKEW_MAX
if (!any(basic_valid)) stop("No numerically valid posterior draws.")
prob_epsilon_nonpositive <- mean(epsilon_gamma[basic_valid] <= 0)
prob_xi_nonpositive <- mean(xi_gamma[basic_valid] <= 0)
prob_both_nonpositive <- mean(epsilon_gamma[basic_valid] <= 0 & xi_gamma[basic_valid] <= 0)
sign_diagnostics <- data.frame(
  numerically_valid_draws = sum(basic_valid),
  P_gamma_epsilon_nonpositive = prob_epsilon_nonpositive,
  P_gamma_xi_nonpositive = prob_xi_nonpositive,
  P_both_nonpositive = prob_both_nonpositive
)
cat("\nUnconstrained native-SN posterior sign probabilities:\n")
print(sign_diagnostics)

summary_frame <- function(draws) {
  z <- t(vapply(as.data.frame(draws), summarize_draws, numeric(5)))
  data.frame(parameter = rownames(z), z, row.names = NULL, check.names = FALSE)
}
# These summaries retain the unrestricted fitted posterior.
unrestricted_betas <- as.data.frame(evaluated[basic_valid, centered_names, drop = FALSE])
```

```r
names(unrestricted_betas) <- beta_names
unrestricted_draws <- data.frame(
  beta0_mean_centered = evaluated[basic_valid, "int_obs"], unrestricted_betas,
  precision_epsilon = epsilon_precision[basic_valid],
  precision_xi = xi_precision[basic_valid],
  gamma_epsilon = epsilon_gamma[basic_valid], gamma_xi = xi_gamma[basic_valid])
unrestricted_summary <- summary_frame(unrestricted_draws)
```

Table 4: Unrestricted posterior sign diagnostics

Item	Value
numerically_valid_draws	12000
P_gamma_epsilon_nonpositive	1
P_gamma_xi_nonpositive	1
P_both_nonpositive	1

4.3 Restore stocastic components and intercepts

```r
# Keep every numerically valid draw; do not filter the main posterior by sign.
all_draw_ids <- which(basic_valid)
joint_draws_all <- evaluated[basic_valid, , drop = FALSE]
D_all <- nrow(joint_draws_all)
B_all <- joint_draws_all[, centered_names, drop = FALSE]
# Signed mean shifts of the uncentered SN errors. This transformation works
# for either skewness sign and does NOT label positive skewness inefficiency.
sn_mean_shift <- function(precision, gamma) {
  delta <- delta_from_gamma(gamma)
  sqrt((1 / precision) / (1 - 2 * delta^2 / pi)) * delta * C_HALF_NORMAL
}
shift_xi_all <- sn_mean_shift(xi_precision[basic_valid], xi_gamma[basic_valid])
shift_epsilon_all <- sn_mean_shift(epsilon_precision[basic_valid], epsilon_gamma[basic_valid])
Xi_centered_draws <- joint_draws_all[, xi_names, drop = FALSE]
Xi_draws <- sweep(Xi_centered_draws, 1L, shift_xi_all, FUN = "+")
colnames(Xi_centered_draws) <- colnames(Xi_draws) <- as.character(farm_ids)
# Rows are joint draws; columns are observations in dat order.
frontier_mean_draws <- cbind(joint_draws_all[, "int_obs"], B_all) %*% t(X)
epsilon_centered_draws <- matrix(dat$y, D_all, NT, byrow = TRUE) -
  frontier_mean_draws - Xi_centered_draws[, dat$id, drop = FALSE]
epsilon_draws <- sweep(epsilon_centered_draws, 1L, shift_epsilon_all, FUN = "+")
colnames(epsilon_draws) <- colnames(epsilon_centered_draws) <- paste0("obs_", seq_len(NT))
rm(frontier_mean_draws)
unrestricted_draws$beta0_mean <- joint_draws_all[, "int_obs"] - as.numeric(B_all %*%
 ↪  input_means)
unrestricted_draws$beta0_SN_location <- unrestricted_draws$beta0_mean - shift_xi_all -
 ↪  shift_epsilon_all
unrestricted_draws$returns_to_scale <- rowSums(B_all)
unrestricted_summary <- summary_frame(unrestricted_draws)

Xi_summary <- cbind(data.frame(farm_id = farm_ids),
  summary_frame(Xi_draws)[, -1L, drop = FALSE])
Xi_centered_summary <- cbind(data.frame(farm_id = farm_ids),
  summary_frame(Xi_centered_draws)[, -1L, drop = FALSE])
Xi_summary$P_Xi_positive <- colMeans(Xi_draws > 0)
```

```r
epsilon_summary <- cbind(dat[c("original_row", "farm_id", "year")],
  summary_frame(epsilon_draws)[, -1L, drop = FALSE])

cat("\nRetained", D_all, "of", nrow(evaluated), "joint draws; NO sign filtering.\n")
print(unrestricted_summary, digits = 5, row.names = FALSE)
```

4.4 Retain draws

```r
production <- prepare_production_draws(
  evaluated = evaluated,
  basic_valid = basic_valid,
  epsilon_gamma = epsilon_gamma,
  xi_gamma = xi_gamma,
  run_production = RUN_PRODUCTION_TE
)

# Use the selected draws in the following calculations.
do_production <- production$do_production
production_draw_ids <- production$production_draw_ids
D <- production$D
evaluated_production <- production$evaluated_production
```

5 Recover frontier parameters and TE

5.1 Component scales, frontier coefficients, and inefficency-to-noise ratios

```r
if (do_production) {
epsilon_map <- map_sn_draws_to_components(
  epsilon_precision[production_draw_ids], epsilon_gamma[production_draw_ids]
)
xi_map <- map_sn_draws_to_components(
  xi_precision[production_draw_ids], xi_gamma[production_draw_ids]
)

beta0_centered <- evaluated_production[, "int_obs"] - xi_map$mean_shift -
 ↪  epsilon_map$mean_shift
B <- evaluated_production[, centered_names, drop = FALSE]
beta0 <- beta0_centered - as.numeric(B %*% input_means)
beta_draws <- as.data.frame(B); names(beta_draws) <- beta_names
structural_draws <- data.frame(
  beta0 = beta0, beta_draws,
  sigma_v0 = xi_map$sigma_gaussian, sigma_u0 = xi_map$sigma_halfnormal,
  sigma_v = epsilon_map$sigma_gaussian, sigma_u = epsilon_map$sigma_halfnormal,
  lambda0 = xi_map$lambda, lambda = epsilon_map$lambda,
  # Ratio within each JOINT posterior draw; not a ratio of posterior means.
  Lambda = ifelse(epsilon_map$sigma_halfnormal > 0,
    xi_map$sigma_halfnormal / epsilon_map$sigma_halfnormal, NA_real_),
  returns_to_scale = rowSums(B),
  gamma_xi = xi_map$gamma, gamma_epsilon = epsilon_map$gamma)
if (anyNA(structural_draws$Lambda)) {
  warning("Lambda is undefined in ", sum(is.na(structural_draws$Lambda)),
```

```
    " draws with zero sigma_u; these remain in memory and are omitted only from Lambda
     ↪  summaries.")
}
structural_summary <- summary_frame(structural_draws)
cat("\nSIGN-CONDITIONED production-frontier parameters; beta0 uses original log/level
 ↪  regressors.\n")
print(structural_summary, digits = 5, row.names = FALSE)
}
```

Table 5: Production-sign-conditioned structural parameters

parameter	mean	sd	2.5%	Median	97.5%
beta0	11.7447	0.0117	11.7215	11.7450	11.7667
beta_cows	0.6421	0.0205	0.6017	0.6422	0.6824
beta_land	0.0348	0.0132	0.0091	0.0347	0.0612
beta_labor	0.0351	0.0171	0.0018	0.0353	0.0676
beta_feed	0.3991	0.0108	0.3784	0.3992	0.4201
sigma_v0	0.0837	0.0060	0.0723	0.0838	0.0952
sigma_u0	0.1315	0.0113	0.1062	0.1318	0.1512
sigma_v	0.0663	0.0020	0.0625	0.0662	0.0700
sigma_u	0.0781	0.0047	0.0680	0.0784	0.0871
lambda0	1.5871	0.2352	1.1156	1.5729	2.0913
lambda	1.1819	0.1022	0.9715	1.1839	1.3941
Lambda	1.6904	0.1843	1.3368	1.6805	2.0207
returns_to_scale	1.1111	0.0186	1.0745	1.1111	1.1479
gamma_xi	-0.3268	0.0750	-0.4786	-0.3242	-0.1730
gamma_epsilon	-0.1949	0.0332	-0.2651	-0.1951	-0.1284

5.2 INLA and FIML comparison

```
comparison <- NULL
fiml_comparison_note <- "No FIML estimates were supplied in FIML_INPUT."
if (do_production && !is.null(FIML_INPUT)) {
  mle <- FIML_INPUT
  if (!is.data.frame(mle) || !all(c("parameter", "estimate") %in% names(mle)) ||
      !is.numeric(mle$estimate)) {
    stop("FIML_INPUT must be a data.frame with parameter and numeric estimate columns.")
  }
  mle <- use_v0_names(mle)
  if (anyNA(mle$parameter) || any(!nzchar(mle$parameter)) ||
      anyDuplicated(mle$parameter)) {
    stop("FIML_INPUT parameter names must be nonempty and unique.")
  }
  if (!("standard_error" %in% names(mle))) mle$standard_error <- NA_real_
  if (!is.numeric(mle$standard_error)) {
    stop("FIML standard_error must be numeric; use NA_real_ when unavailable.")
  }
  available <- is.finite(mle$estimate)
  if (!any(available)) {
    fiml_comparison_note <- paste(
      "FIML comparison omitted: FIML_INPUT contains no finite point estimates.",
      "Replace the NA placeholders with actual FIML estimates and knit again."
```

```
    )
    warning(fiml_comparison_note, call. = FALSE)
  } else {
    if (any(!available)) {
      warning("FIML comparison omits missing estimates for: ",
        paste(mle$parameter[!available], collapse = ", "), call. = FALSE)
    }
    mle <- mle[available, , drop = FALSE]
    if (any(mle$standard_error < 0, na.rm = TRUE)) {
      stop("FIML standard errors must be nonnegative.")
    }
    mle$standard_error[!is.finite(mle$standard_error)] <- NA_real_
    names(mle)[names(mle) == "estimate"] <- "FIML_estimate"
    names(mle)[names(mle) == "standard_error"] <- "FIML_standard_error"
    comparison <- merge(mle, structural_summary, by = "parameter", sort = FALSE)
    # Preserve the order of parameters in the embedded FIML input.
    comparison <- comparison[order(match(comparison$parameter, mle$parameter)), , drop =
↪   FALSE]
    if (!nrow(comparison)) {
      comparison <- NULL
      fiml_comparison_note <- paste(
        "No FIML parameter names match the INLA parameter names.",
        "Check beta0, beta_cows, beta_land, beta_labor, beta_feed,",
        "sigma_v0, sigma_u0, sigma_v, and sigma_u."
      )
      warning(fiml_comparison_note, call. = FALSE)
    } else {
      comparison$INLA_minus_FIML <- comparison$mean - comparison$FIML_estimate
      fiml_comparison_note <- ""
    }
  }
}
```

Table 6: INLA and FIML comparison

parameter	FIML estimate	FIML SE	INLA mean	INLA SD	Difference
beta0	11.7609	0.0167	11.7447	0.0117	-0.0162
beta_cows	0.6407	0.0205	0.6421	0.0205	0.0014
beta_land	0.0357	0.0133	0.0348	0.0132	-0.0009
beta_labor	0.0346	0.0170	0.0351	0.0171	0.0005
beta_feed	0.3990	0.0108	0.3991	0.0108	0.0001
sigma_v0	0.0742	0.0112	0.0837	0.0060	0.0095
sigma_u0	0.1467	0.0191	0.1315	0.0113	-0.0152
sigma_v	0.0633	0.0038	0.0663	0.0020	0.0030
sigma_u	0.0842	0.0083	0.0781	0.0047	-0.0061

FIML source: Dataset 1 robustness-check table in `parameter.docx`; reported estimates and standard errors are embedded in this Rmd.

5.3 Allocate and generate efficiency draws

```r
if (do_production) {
set.seed(BASE_SEED + 100L)
xi_c_draws <- evaluated_production[, xi_names, drop = FALSE]
TE_persistent_draws <- matrix(NA_real_, N, D)
TE_transient_draws <- matrix(NA_real_, NT, D)
TE_overall_draws <- matrix(NA_real_, NT, D)
persistent_te_sum <- numeric(N)
transient_te_sum <- numeric(NT)
overall_te_sum <- numeric(NT)
}
```

```r
if (do_production) {
for (d in seq_len(D)) {
  xi_raw <- xi_c_draws[d, ] + xi_map$mean_shift[d]
  p0 <- conditional_halfnormal_parameters(xi_raw, xi_map$sigma_gaussian[d],
                                          xi_map$sigma_halfnormal[d])
  u0 <- rtruncnorm_lower0(N, p0$mean, p0$sd)
  # Use centered coefficients and centered frontier intercept together.
  frontier <- as.numeric(X %*% c(beta0_centered[d], B[d, ]))
  epsilon_raw <- dat$y - frontier - xi_raw[dat$id]
  pt <- conditional_halfnormal_parameters(epsilon_raw, epsilon_map$sigma_gaussian[d],
                                          epsilon_map$sigma_halfnormal[d])
  u <- rtruncnorm_lower0(NT, pt$mean, pt$sd)
  # Exact conditional expectations, as in the uploaded MC code.
  te0_mean <- conditional_halfnormal_te(xi_raw, xi_map$sigma_gaussian[d],
 ↪  xi_map$sigma_halfnormal[d])
  tet_mean <- conditional_halfnormal_te(epsilon_raw, epsilon_map$sigma_gaussian[d],
 ↪  epsilon_map$sigma_halfnormal[d])
  teo_mean <- te0_mean[dat$id] * tet_mean
  if (any(!is.finite(c(te0_mean, tet_mean, teo_mean)))) stop("Nonfinite conditional TE
   ↪  means.")
  persistent_te_sum <- persistent_te_sum + te0_mean
  transient_te_sum <- transient_te_sum + tet_mean
  overall_te_sum <- overall_te_sum + teo_mean
  TE_persistent_draws[, d] <- exp(-u0)
  TE_transient_draws[, d] <- exp(-u)
  TE_overall_draws[, d] <- exp(-u0[dat$id] - u)
  if (d %% 250L == 0L || d == D) cat("TE draws processed:", d, "/", D, "\n")
}
}
```

5.4 Individual summaries and ranking

```r
if (do_production) {
te_summary <- function(m, prefix) {
  z <- data.frame(
    mean = rowMeans(m), sd = apply(m, 1L, sd),
    median = apply(m, 1L, median),
    lower = apply(m, 1L, quantile, probs = 0.025),
    upper = apply(m, 1L, quantile, probs = 0.975)
  )
  names(z) <- paste0(prefix, "_", names(z))
  z
```

```
}

# STEP 8. Construct persistent and farm-year efficiency tables
persistent_results <- cbind(data.frame(farm_id = farm_ids),
                            te_summary(TE_persistent_draws, "TE_persistent"))
persistent_results$TE_persistent_mean <- persistent_te_sum / D
# Rank 1 is the highest posterior mean efficiency; ties receive equal ranks.
persistent_results$rank <- rank(-persistent_results$TE_persistent_mean, ties.method = "min")
persistent_results <- persistent_results[order(persistent_results$rank), ]
farm_year_results <- cbind(
  dat[c("original_row", "farm_id", "year", "log_output", regressor_names)],
  te_summary(TE_persistent_draws[dat$id, , drop = FALSE], "TE_persistent"),
  te_summary(TE_transient_draws, "TE_transient"),
  te_summary(TE_overall_draws, "TE_overall")
)
farm_year_results$TE_persistent_mean <- (persistent_te_sum / D)[dat$id]
farm_year_results$TE_transient_mean <- transient_te_sum / D
farm_year_results$TE_overall_mean <- overall_te_sum / D
# Means above use the original analytical conditional predictors.
# SDs/intervals use random conditional inefficiency draws and include
# inefficiency uncertainty; they are not intervals of conditional means.
# Average TE is computed within each JOINT draw, then summarized.
}
```

5.5 Aggregate and yearly efficiency

```
if (do_production) {
aggregate_TE_draws <- data.frame(
  persistent = colMeans(TE_persistent_draws),
  transient = colMeans(TE_transient_draws),
  overall = colMeans(TE_overall_draws)
)
aggregate_TE_summary <- summary_frame(aggregate_TE_draws)
aggregate_TE_summary$mean <- c(mean(persistent_te_sum/D), mean(transient_te_sum/D),
 ↪  mean(overall_te_sum/D))
# Ranges across individual posterior mean TE estimates (different from the
# credible intervals for average TE already in q0.025/q0.975).
te_mean_values <- list(
  persistent = persistent_results$TE_persistent_mean,
  transient = farm_year_results$TE_transient_mean,
  overall = farm_year_results$TE_overall_mean
)
aggregate_TE_summary$min <- vapply(te_mean_values[aggregate_TE_summary$parameter], min,
 ↪  numeric(1))
aggregate_TE_summary$max <- vapply(te_mean_values[aggregate_TE_summary$parameter], max,
 ↪  numeric(1))
# Persistent average gives each farm equal weight; the other averages give
# each observation equal weight. These coincide for this balanced dataset.
}
```

Table 7: Average technical efficiency, conditional on production signs

parameter	mean	sd	2.5%	Median	97.5%	min	max
persistent	0.9030	0.0085	0.8875	0.9027	0.9199	0.7247	0.9752
transient	0.9411	0.0034	0.9351	0.9409	0.9479	0.7385	0.9874
overall	0.8498	0.0084	0.8341	0.8496	0.8658	0.6193	0.9569

```
if (do_production) {
year_TE_summary <- do.call(rbind, lapply(sort(unique(dat$year)), function(yr) {
  rows <- which(dat$year == yr)
  draw_means <- data.frame(
    persistent = colMeans(TE_persistent_draws[dat$id[rows], , drop = FALSE]),
    transient = colMeans(TE_transient_draws[rows, , drop = FALSE]),
    overall = colMeans(TE_overall_draws[rows, , drop = FALSE])
  )
  cbind(year = yr, summary_frame(draw_means))
}))
# Match the same analytical posterior-mean predictors in yearly summaries.
for (yr in sort(unique(dat$year))) {
  rows <- which(dat$year == yr)
  sel <- which(year_TE_summary$year == yr)
  year_TE_summary$mean[sel] <- c(
    mean((persistent_te_sum/D)[dat$id[rows]]),
    mean((transient_te_sum/D)[rows]), mean((overall_te_sum/D)[rows])
  )
}
}
```

Table 8: Average efficiency by year

Year	Component	Mean	SD	2.5%	97.5%
1993	persistent	0.9030	0.0085	0.8875	0.9199
1993	transient	0.9305	0.0050	0.9213	0.9402
1993	overall	0.8402	0.0089	0.8234	0.8570
1994	persistent	0.9030	0.0085	0.8875	0.9199
1994	transient	0.9385	0.0043	0.9306	0.9467
1994	overall	0.8475	0.0087	0.8310	0.8640
1995	persistent	0.9030	0.0085	0.8875	0.9199
1995	transient	0.9442	0.0037	0.9372	0.9512
1995	overall	0.8526	0.0085	0.8366	0.8690
1996	persistent	0.9030	0.0085	0.8875	0.9199
1996	transient	0.9443	0.0037	0.9374	0.9513
1996	overall	0.8528	0.0085	0.8368	0.8692
1997	persistent	0.9030	0.0085	0.8875	0.9199
1997	transient	0.9431	0.0038	0.9360	0.9504
1997	overall	0.8515	0.0085	0.8355	0.8678
1998	persistent	0.9030	0.0085	0.8875	0.9199
1998	transient	0.9458	0.0035	0.9392	0.9526
1998	overall	0.8541	0.0085	0.8380	0.8703

5.6 Production diagnostics

```
if (do_production) {
production_post_seconds <- proc.time()[["elapsed"]] - production$production_post_start
production_diagnostics <- data.frame(
  requested_draws = POSTERIOR_DRAWS, valid_draws = D,
  excluded_draws = POSTERIOR_DRAWS - D,
  prob_epsilon_nonpositive = prob_epsilon_nonpositive,
  prob_xi_nonpositive = prob_xi_nonpositive,
  prob_both_nonpositive = prob_both_nonpositive,
  production_sign_conditioned = TRUE,
  fit_seconds = fit_seconds, postprocess_seconds = production_post_seconds,
  carrier_precision = CARRIER_PRECISION,
  PC_composed_SD_upper = PC_COMPONENT_SD_UPPER,
  PC_composed_SD_tail_probability = PC_COMPONENT_SD_PROB,
  PC_skew_lambda = PC_SKEW_PARAM
)
}
```

```
post_seconds <- proc.time()[["elapsed"]] - post_start
has_production <- do_production
diagnostics <- data.frame(
  requested_draws = POSTERIOR_DRAWS,
  numerically_valid_draws = D_all,
  numerically_invalid_draws = sum(!basic_valid),
  production_compatible_draws = D,
  P_both_nonpositive = prob_both_nonpositive,
  sign_filter_applied_to_main_results = FALSE,
  fit_seconds = fit_seconds,
  postprocess_seconds = post_seconds
)
```

Table 9: Main computation diagnostics

Item	Value
requested_draws	12000
numerically_valid_draws	12000
numerically_invalid_draws	0
production_compatible_draws	12000
P_both_nonpositive	1
sign_filter_applied_to_main_results	FALSE
fit_seconds	1.93
postprocess_seconds	37.13

Table 10: Production post-processing diagnostics

Item	Value
requested_draws	12000
valid_draws	12000
excluded_draws	0
prob_epsilon_nonpositive	1
prob_xi_nonpositive	1
prob_both_nonpositive	1

production_sign_conditioned	TRUE
fit_seconds	1.93
postprocess_seconds	24.57
carrier_precision	1e-06
PC_composed_SD_upper	1
PC_composed_SD_tail_probability	0.01
PC_skew_lambda	10

6 Efficiency figures

```
p_TE_base <- p_year_base <- NULL
if (has_production && MAKE_PLOTS) {
  suppressPackageStartupMessages(library(ggplot2))
TE_plot_data <- rbind(
    data.frame(TE = persistent_results$TE_persistent_mean, component = "Persistent TE"),
    data.frame(TE = farm_year_results$TE_transient_mean, component = "Transient TE"),
    data.frame(TE = farm_year_results$TE_overall_mean, component = "Overall TE")
  )
  TE_plot_data$component <- factor(
    TE_plot_data$component,
    levels = c("Persistent TE", "Transient TE", "Overall TE")
  )
  if (any(!is.finite(TE_plot_data$TE)) || any(TE_plot_data$TE < 0 | TE_plot_data$TE > 1)) {
    stop("Plotting requires finite posterior mean TE values between zero and one.")
  }

  # 13b. Build three histogram panels with genuinely free x and y scales.
  # bins=30 allows each panel to choose its own breaks from its own range.
  # Fixed breaks spanning 0 to 1 would force the panels back to that range.
  p_TE_base <- ggplot(TE_plot_data, aes(x = TE, fill = component)) +
    geom_histogram(bins = 30, colour = "white", linewidth = 0.3) +
    facet_wrap(~ component, nrow = 1, scales = "free") +
    scale_fill_manual(values = c(
      "Persistent TE" = "#3C8DAD", "Transient TE" = "#57A773", "Overall TE" = "#CB8A45"
    )) +
    labs(x = NULL, y = NULL) +
    theme_minimal(base_size = 13) +
    theme(
      legend.position = "none", panel.grid.minor = element_blank(),
      strip.text = element_text(face = "bold", size = 13),
      panel.spacing = grid::unit(1, "lines")
    )

year_plot_data <- year_TE_summary
  year_plot_data$component <- factor(
    year_plot_data$parameter,
    levels = c("persistent", "transient", "overall"),
    labels = c("Persistent TE", "Transient TE", "Overall TE")
  )
  year_plot_data <- year_plot_data[
    order(year_plot_data$component, year_plot_data$year), , drop = FALSE
  ]

  # 13e. Plot posterior means and 95% credible intervals; no overall title.
```

```r
  # The full 0-to-1 TE range keeps credible intervals visible.
  p_year_base <- ggplot(
    year_plot_data, aes(x = year, y = mean, colour = component, group = component)
  ) +
    geom_errorbar(aes(ymin = q0.025, ymax = q0.975), width = 0.12, linewidth = 0.5) +
    geom_line(linewidth = 0.8) +
    geom_point(aes(shape = component), size = 2.5) +
    scale_colour_manual(values = c(
      "Persistent TE" = "#3C8DAD", "Transient TE" = "#57A773", "Overall TE" = "#CB8A45"
    )) +
    scale_shape_manual(values = c("Persistent TE" = 16, "Transient TE" = 17, "Overall TE" =
    ↪  18)) +
    scale_x_continuous(breaks = sort(unique(year_plot_data$year))) +
    coord_cartesian(ylim = c(0, 1)) +
    labs(x = "Year", y = "Average TE", colour = "Component", shape = "Component") +
    theme_minimal(base_size = 13) +
    theme(legend.position = "bottom", panel.grid.minor = element_blank())
}
```

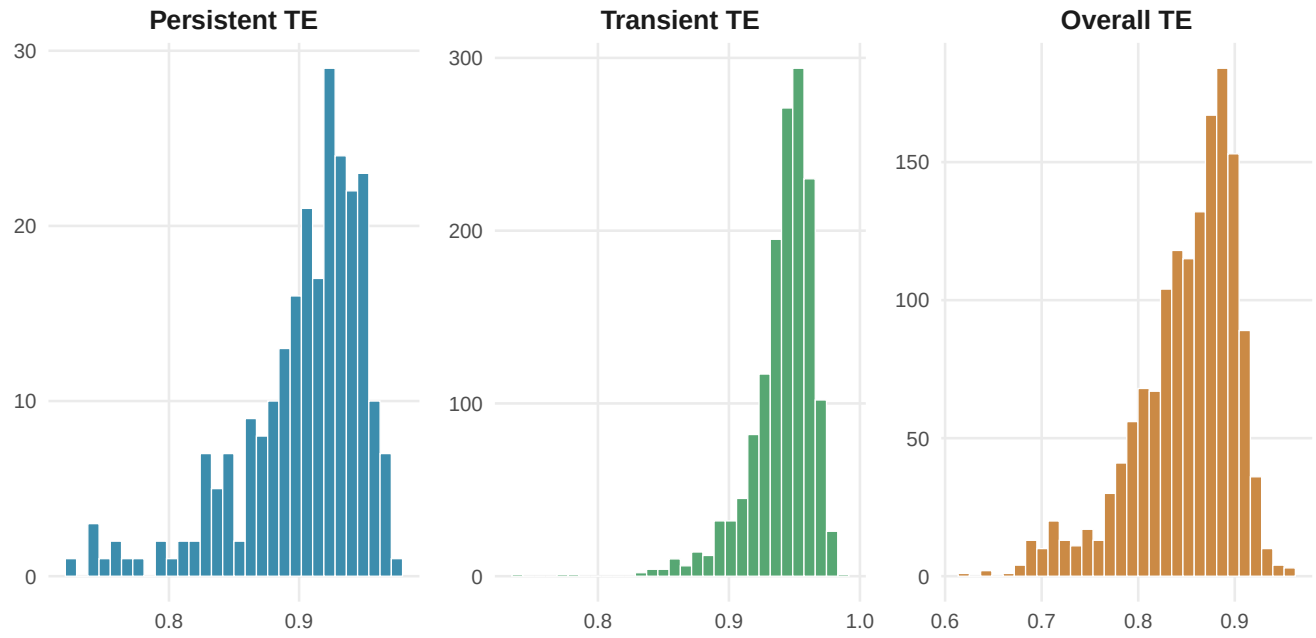

Figure 1: Distributions of individual posterior mean efficiencies, conditional on production signs.

7 Reproducibility

7.1 Software environment

```
#> R version 4.5.3 (2026-03-11 ucrt)
#> Platform: x86_64-w64-mingw32/x64
#> Running under: Windows 11 x64 (build 26100)
#>
#> Matrix products: default
#>   LAPACK version 3.12.1
```

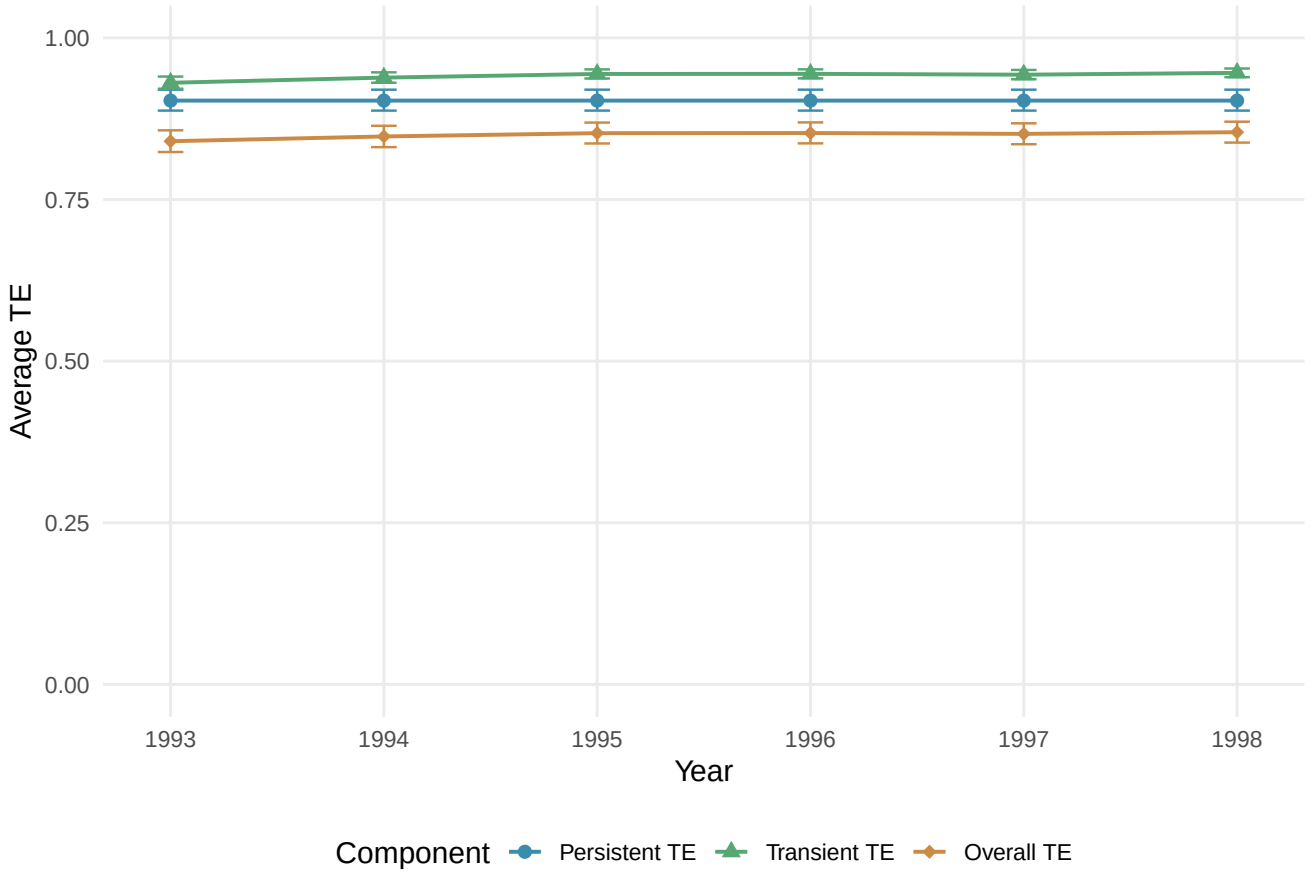

Figure 2: Annual average technical efficiency and 95 percent credible intervals, conditional on production signs.

```
#>
#> locale:
#> [1] LC_COLLATE=Chinese (Simplified)_China.utf8
#> [2] LC_CTYPE=Chinese (Simplified)_China.utf8
#> [3] LC_MONETARY=Chinese (Simplified)_China.utf8
#> [4] LC_NUMERIC=C
#> [5] LC_TIME=Chinese (Simplified)_China.utf8
#>
#> time zone: Asia/Shanghai
#> tzcode source: internal
#>
#> attached base packages:
#> [1] stats     graphics  grDevices utils     datasets  methods   base
#>
#> other attached packages:
#> [1] ggplot2_4.0.2 INLA_25.10.19 Matrix_1.7-4
#>
#> loaded via a namespace (and not attached):
#>  [1] gtable_0.3.6          xfun_0.57             collapse_2.1.7
#>  [4] lattice_0.22-9        numDeriv_2016.8-1.1   vctrs_0.7.2
#>  [7] tools_4.5.3           Rdpack_2.6.6          generics_0.1.4
#> [10] stats4_4.5.3          parallel_4.5.3        sandwich_3.1-1
#> [13] proxy_0.4-29          tibble_3.3.1          ucminf_1.2.3
#> [16] pkgconfig_2.0.3       KernSmooth_2.23-26    hpa_1.3.4
#> [19] mnorm_1.2.3           RColorBrewer_1.1-3    S7_0.2.1
#> [22] RcppParallel_5.1.11-2 lifecycle_1.0.5       cubature_2.1.4-1
#> [25] compiler_4.5.3        farver_2.1.2          fmesher_0.7.0
#> [28] MatrixModels_0.5-4    maxLik_1.5-2.2        mnormt_2.1.2
#> [31] fastGHQuad_1.0.1      codetools_0.2-20      class_7.3-23
#> [34] htmltools_0.5.9       yaml_2.3.12           Formula_1.2-5
#> [37] pillar_1.11.1         MASS_7.3-65           classInt_0.4-11
#> [40] randtoolbox_2.0.5     plm_2.6-7             iterators_1.0.14
#> [43] foreach_1.5.2         nlme_3.1-168          tidyselect_1.2.1
#> [46] bdsmatrix_1.3-7       digest_0.6.39         sf_1.1-0
#> [49] dplyr_1.2.0           labeling_0.4.3        splines_4.5.3
#> [52] miscTools_0.6-30      fastmap_1.2.0         grid_4.5.3
#> [55] cli_3.6.6             magrittr_2.0.4        qrng_0.0-11
#> [58] dichromat_2.0-0.1     e1071_1.7-17          sn_2.1.3
#> [61] withr_3.0.2           scales_1.4.0          rmarkdown_2.30
#> [64] httr_1.4.8            otel_0.2.0            rngWELL_0.10-10
#> [67] zoo_1.8-15            evaluate_1.0.5        knitr_1.51
#> [70] rbibutils_2.4.1       lmtest_0.9-40         doParallel_1.0.17
#> [73] trustOptim_0.8.7.4    nleqslv_3.3.7         rlang_1.2.0
#> [76] Rcpp_1.1.1-1          glue_1.8.0            DBI_1.3.0
#> [79] rstudioapi_0.18.0     R6_2.6.1              sfaR_1.0.1
#> [82] units_1.0-1           texreg_1.39.5         marqLevAlg_2.0.8
```