	Gradient	**Subgrad**	**Prox grad**	**Stochastic prox grad**
Criterion	smooth f	any f	smooth + simple, $f = g + h$	smooth + simple, $f = g + h$
Constraints	projection onto constraint set	projection onto constraint set	constrained prox operator	constrained prox operator
Opti parameters	fixed step size ($t \leq 1/L$) or use line search	diminishing step sizes	fixed step size ($t \leq 1/L$) or use line search	fixed step size, mini-batch size
Iteration cost	cheap (compute gradient)	cheap (compute subgradient)	moderately cheap (evaluate prox)	very cheap (compute stochastic gradient, evaluate prox)
Rate	$O(\frac{1}{\epsilon})$ (acceleration: $O(\frac{1}{\sqrt{\epsilon}})$, strong convexity: $O(\log(\frac{1}{\epsilon}))$)	$O(\frac{1}{\epsilon^2})$	$O(\frac{1}{\epsilon})$ (acceleration: $O(\frac{1}{\sqrt{\epsilon}})$, strong convexity: $O(\log(\frac{1}{\epsilon}))$)	$O(1/\epsilon^2)$, but practically converges rapidly at the start

	Newton	**Barrier method**	**Primal-dual int-point**	**Prox Newton**
Criterion	doubly smooth f	doubly smooth f	doubly smooth f	dbl smooth + simple, $f = g + h$
Constraints	equality constraints	equality, dbl smooth h_i (inequality constraints)	equality, dbl smooth h_i (inequality constraints)	constrained H-prox
Opti parameters	pure step size ($t = 1$) or use line search	inner: pure step size or line search; outer: barrier update factor	line search for step size, barrier update factor	pure step size or use line search
Iteration cost	moderate to expensive (compute Hessian and solve linear system)	expensive to very expensive (one iter solves one smoothed problem, by Newton)	moderate to expensive (one iter performs one Newton step)	expensive to very expensive (evaluate H-prox)
Rate	$O(\log\log(\frac{1}{\epsilon}))$ (local quadratic rate)	$O(\log(\frac{1}{\epsilon}))$ (both iters, and Newton steps)	$O(\log(\frac{1}{\epsilon}))$	$O(\log\log(\frac{1}{\epsilon}))$ (local quadratic rate)