A FULL RESULTS ON TSP AND CVRP (SEEN SCALES)

In Table 1, regarding the learning-oriented methods, we mainly displayed their results yielded with the data augmentation strategy (A). Here we show the full results with (A) and without (N) the data augmentation in Table 5. As shown, the inference time without data augmentation is significantly shorter than that with data augmentation. On the other hand, it is clear that both Ours-inter and Ours-intra achieve the best average performance among all deep models, no matter the data augmentation strategy is used or not.

Table 5: Comparison results on TSP and CVRP (seen scales).

	Method	Test on N=60			Test on N=100			Test on N=150			Average of
		Obj.	Gap	Time$^{\#}$	Obj.	Gap	Time$^{\#}$	Obj.	Gap	Time$^{\#}$	Total costs
TSP	Concorde	6.1729	-	(7m)	7.7646	-	(1.7h)	9.3462	-	(22m)	7.7612
	LKH3	6.1729	0.00%	(14m)	7.7646	0.00%	(9.8h)	9.3462	0.00%	(2.1h)	7.7612
	(N) AMDKD-POMO*	6.2115	0.62%	6s	7.8312	0.86%	15s	9.5229	1.89%	4s	7.8552
	(N) POMO-60	6.1811	0.13%	~	7.8514	1.12%	~	9.7019	3.81%	~	7.9115
	(N) POMO-100	6.1972	0.39%	~	7.7908	0.34%	~	9.4500	1.11%	~	7.8160
	(N) POMO-150	6.3031	2.11%	~	7.8702	1.36%	~	9.3806	0.39%	~	7.8513
	(N) AMDKD-POMO	6.2076	0.56%	~	7.8292	0.83%	~	9.5035	1.68%	~	7.8468
	(N) Omni-POMO‡	6.3065	2.16%	5s	7.9425	2.29%	18s	9.5795	2.50%	5s	7.9428
	(N) Ours-intra	6.1882	0.25%	6s	7.8007	0.47%	15s	9.4275	0.87%	4s	7.8055
	(N) Ours-inter	6.1889	0.26%	~	7.7995	0.45%	~	9.4010	0.59%	~	7.7965
	(A) AMDKD-POMO*	6.1828	0.16%	36s	7.7930	0.37%	2m	9.4539	1.15%	33s	7.8092
	(A) POMO-60	**6.1746**	**0.03%**	~	7.8050	0.52%	~	9.5909	2.62%	~	7.8568
	(A) POMO-100	6.1768	0.06%	~	**7.7753**	**0.14%**	~	9.3987	0.56%	~	7.7836
	(A) POMO-150	6.1928	0.32%	~	7.7875	0.30%	~	**9.3812**	**0.36%**	~	7.7868
	(A) AMDKD-POMO	6.1820	0.15%	~	7.7916	0.35%	~	9.4473	1.08%	~	7.8070
	(A) Omni-POMO‡	6.2351	1.01%	34s	7.8650	1.29%	2.5m	9.4958	1.60%	37s	7.8653
	(A) Ours-inter	*6.1758*	*0.05%*	36s	7.7775	0.17%	2m	9.3883	0.45%	33s	*7.7805*
	(A) Ours-intra	6.1758	0.05%	~	*7.7764*	*0.15%*	~	*9.3820*	*0.38%*	~	**7.7781**
CVRP	HGS	11.9471	-	(15.3h)	15.5642	-	(25.6h)	19.0554	-	(6.2h)	15.5222
	LKH3	11.9694	0.19%	(3.5d)	15.6473	0.53%	(6.5d)	19.2208	0.87%	(13h)	15.6125
	(N) AMDKD-POMO*	12.5745	5.25%	7s	15.9485	2.47%	22s	20.0788	5.37%	6s	16.2006
	(N) POMO-60	12.1757	1.91%	~	16.3058	4.77%	~	20.5332	7.76%	~	16.3382
	(N) POMO-100	12.4140	3.91%	~	15.8369	1.75%	~	19.8641	4.24%	~	16.0383
	(N) POMO-150	12.7088	6.38%	~	16.0346	3.02%	~	19.4822	2.24%	~	16.0752
	(N) AMDKD-POMO	12.2731	2.73%	~	15.9498	2.48%	~	19.6619	3.18%	~	15.9616
	(N) Omni-POMO‡	12.4879	4.53%	6s	16.1511	3.77%	20s	19.7527	3.66%	6s	16.1505
	(N) Ours-intra	12.1818	1.97%	7s	15.9166	2.26%	22s	19.5555	2.62%	6s	15.8846
	(N) Ours-inter	12.1838	1.98%	~	15.9093	2.22%	~	19.5342	2.51%	~	15.8758
	(A) AMDKD-POMO*	12.3561	3.42%	56s	15.8854	2.06%	3m	19.8395	4.12%	33s	16.0270
	(A) POMO-60	**12.0656**	**0.99%**	~	16.0914	3.39%	~	20.2573	6.31%	~	16.1381
	(A) POMO-100	12.2531	2.56%	~	**15.7544**	**1.22%**	~	19.6856	3.31%	~	15.8977
	(A) POMO-150	12.4322	4.06%	~	15.8924	2.11%	~	**19.3683**	**1.64%**	~	15.8976
	(A) AMDKD-POMO	12.1487	1.69%	~	15.8119	1.72%	~	19.5280	2.48%	~	15.8362
	(A) Omni-POMO‡	12.2996	2.95%	45s	15.9878	2.72%	2.5m	19.5975	2.85%	45s	15.9616
	(A) Ours-inter	*12.0660*	*1.00%*	56s	15.7848	1.42%	3m	19.4109	1.87%	33s	*15.7539*
	(A) Ours-intra	12.0663	1.00%	~	*15.7781*	*1.37%*	~	*19.3938*	*1.78%*	~	**15.7461**

Bold and ***italics*** refer to the best and the second-best performance, respectively, among all deep models.
~ The inference time of a method equals to that of the preceding method in the above row.
‡ The training size range of Omni-POMO is [50, 200], which is broader than that of ours, i.e., [60, 150].

B DETAILED GENERALIZATION RESULTS ON BENCHMARK DATASETS.

We evaluate all methods on the classic benchmark datasets, including TSPLIB (Reinelt, 1991) and CVRPLIB (Uchoa et al., 2017), in which we choose representative instances with size $N \in [50, 500]$. Note that both benchmark datasets encompass a diverse range of sizes and distributions. The detailed results are shown in Table 6 and Table 7, where the gaps are calculated based on the optimal solution values for the instances. From two tables, we observe that our approach outperforms AMDKD-POMO and all POMO models trained on specific sizes on both benchmark datasets. Note that Omni-POMO is explicitly specialized for improving generalization across both size and distribution, which has the potential to achieve desirable performance on benchmark datasets whose instances encompass a diverse range of sizes and distributions. More specific, the setting that explicitly training with more diverse sizes and distributions, will inherently favor Omni-POMO over our approach. Whereas, our approach still yields competitive performance in comparison against Omni-POMO, e.g., with the average gap of 5.45% (Ours) VS. 5.83% (Omni-POMO) on CVRPLIB, which further underscores the effectiveness of our approach.

Table 6: Detailed generalization results on instances from TSPLIB.

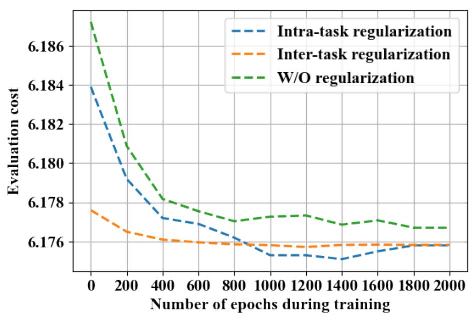
		POMO-60		POMO-100		POMO-150		AMDKD-POMO		Omni-POMO‡		Ours	
Instance	Opt.	Obj.	Gap	Obj.	Gap	Obj.	Gap	Obj.	Gap	Obj.	Gap	Obj.	Gap
berlin52	7544	7544	0.00%	7545	0.01%	7613	0.92%	7545	0.01%	8003	6.08%	7544	0.00%
st70	675	677	0.30%	677	0.30%	678	0.44%	677	0.30%	680	0.74%	677	0.30%
eil76	538	547	1.67%	544	1.12%	549	2.05%	550	3.77%	557	3.53%	544	1.12%
rd100	7910	7920	0.13%	7910	0.00%	7952	0.53%	7934	0.30%	7958	0.61%	7910	0.00%
KroA100	21282	21786	2.34%	21667	1.81%	21596	1.48%	22077	3.74%	21305	0.11%	21738	2.14%
KroB100	22141	22941	3.61%	22370	1.03%	22575	1.96%	22745	2.73%	22650	2.30%	22640	2.25%
lin105	14379	14808	2.98%	14557	1.24%	14808	2.98%	14898	3.61%	14819	3.06%	14753	2.60%
pr124	59030	59031	0.00%	59388	0.61%	59595	0.96%	59521	0.83%	59238	0.35%	59164	0.23%
ch130	6110	6188	1.28%	6133	0.38%	6142	0.52%	6159	0.80%	6251	2.31%	6119	0.15%
pr136	96772	100459	38.12%	97540	0.80%	97668	0.93%	97951	1.22%	97780	1.04%	97258	0.50%
gr137	699	746	6.72%	755	8.01%	759	8.58%	773	10.59%	772	10.44%	747	6.87%
ch150	6528	6679	2.31%	6559	0.48%	6579	0.78%	6583	0.82%	6586	0.89%	6559	0.48%
KroA200	29368	31819	8.35%	30415	3.57%	30015	2.20%	30672	4.44%	29823	1.55%	29951	1.99%
KroB200	29437	32020	8.78%	30880	4.90%	30172	2.50%	30990	5.28%	29814	1.28%	30792	4.60%
ts225	126643	135704	7.16%	130990	3.43%	128045	1.14%	128911	1.79%	128770	1.68%	129297	2.10%
a280	2579	3101	20.24%	2951	0.45%	2788	8.10%	2809	8.92%	2695	4.50%	2828	9.65%
rd400	15281	18055	18.15%	17342	13.49%	16155	5.72%	16160	5.75%	15948	4.37%	15968	4.50%
fl417	11861	14319	20.72%	14396	21.37%	14225	19.93%	14004	18.07%	12683	6.93%	13932	17.46%
pcb442	50778	71914	41.62%	62145	22.39%	54683	7.69%	63643	25.34%	59761	17.69%	61152	20.43%
Avg. Gap	0.00%	-	9.71%	-	4.49%	-	4.18%	-	5.17%	-	3.11%	-	4.07%

Table 7: Detailed generalization results on instances from CVRPLIB.

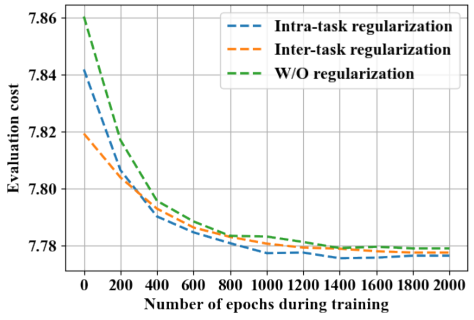
		POMO-60		POMO-100		POMO-150		AMDKD-POMO		Omni-POMO‡		Ours	
Instance	Opt.	Obj.	Gap	Obj.	Gap	Obj.	Gap	Obj.	Gap	Obj.	Gap	Obj.	Gap
A-n53-k7	1010	1315	30.20%	1318	30.50%	1152	14.06%	1111	10.00%	1105	9.41%	1136	12.48%
A-n60-k9	1354	1741	28.58%	1739	28.43%	1657	22.38%	1574	16.25%	1465	8.20%	1453	7.31%
A-n80-k10	1763	2692	52.69%	2740	55.42%	2816	59.73%	2136	21.16%	2127	20.65%	2100	19.12%
X-n101-k25	27591	29786	7.96%	29287	6.15%	29398	6.55%	29306	6.22%	29442	6.71%	28533	3.41%
X-n110-k13	14971	15530	3.73%	15161	1.27%	15130	1.06%	15202	1.54%	15285	2.10%	15102	0.88%
X-n120-k6	13332	14239	6.80%	14570	9.29%	14060	5.46%	14010	5.09%	13944	4.59%	13882	4.13%
X-n129-k18	28940	30154	4.20%	29569	2.17%	29343	1.39%	29702	2.63%	29975	3.58%	29306	1.27%
X-n139-k10	13590	14269	5.00%	14080	3.61%	13855	1.95%	13890	2.21%	14019	3.16%	13812	1.63%
X-n148-k46	43448	46146	6.21%	47621	9.61%	45850	5.53%	46451	6.91%	46438	6.88%	45600	4.95%
X-n157-k13	16876	17663	4.66%	18302	8.45%	18340	8.68%	17523	3.83%	17107	1.37%	17414	3.19%
X-n167-k10	20557	22306	8.51%	21297	3.60%	20995	2.13%	21068	2.49%	21436	4.28%	20960	1.96%
X-n190-k8	16980	18757	10.47%	18164	6.97%	18140	6.83%	18169	7.00%	17645	3.92%	17770	4.65%
X-n200-k36	58578	62737	7.10%	61933	5.73%	61397	4.81%	61384	4.79%	61496	4.98%	61514	5.01%
X-n251-k28	38684	42430	9.69%	41360	6.92%	40024	3.47%	40595	4.94%	40059	3.55%	40046	3.52%
X-n298-k31	34231	38749	13.20%	38611	12.80%	35663	4.18%	37221	8.74%	36384	6.29%	35779	4.52%
X-n351-k40	25896	30289	16.96%	28343	9.45%	27952	7.94%	28315	9.34%	27515	6.25%	27487	6.14%
X-n401-k29	66154	72209	9.15%	71173	7.59%	69641	5.27%	69227	4.65%	68234	3.14%	69682	5.33%
X-n449-k29	55233	63569	15.09%	61915	12.08%	58418	5.77%	59929	8.50%	58037	5.08%	58563	6.03%
X-n491-k59	66483	78463	18.02%	75620	13.74%	71668	7.80%	72087	8.43%	70923	6.68%	71787	7.98%
Avg. Gap	0.00%	-	13.59%	-	12.30%	-	9.21%	-	7.09%	-	5.83%	-	5.45%

C Ablation study on regularization schemes

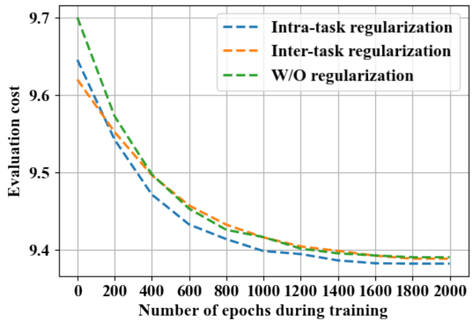
To verify the effectiveness of the regularization schemes designed in our approach, we further analyze the evaluation results of our accomplished models after training on each size with inter-task regularization scheme, intra-task regularization scheme, and without regularization scheme, across three sizes (i.e., 60, 100 and 150) for TSP. The corresponding curves are depicted in Figure 3, where we also present the average objective values across those sizes, in order to show the overall improved cross-size generalization performance. As revealed, both inter-task and intra-task regularization schemes accelerate the learning efficiency and boost the overall results in comparison with the one without regularization scheme, which showcases the effectiveness of the proposed regularization schemes. Moreover, inter-task regularization scheme is superior to the intra-task one in terms of the learning efficiency on smaller sizes (i.e., 60), while the other way round on larger sizes (i.e., 100 and 150). This is reasonable since the intra-task regularization scheme concentrate more on efficiently learning the latest lager sizes. Furthermore, the intra-task achieves the fastest learning efficiency and the lowest objective value when averaging the objective values across all three sizes.



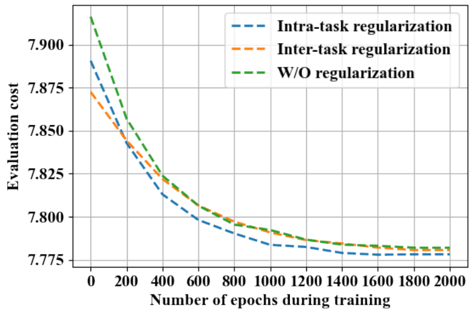
(a) TSP60



(b) TSP100



(c) TSP150



(d) Average on three sizes

Figure 3: Curves of learning progress for our approach on TSP.

Table 8: Validation results on seen and unseen sizes.

	Method	Test on N=100 Obj.	Gap	Test on N=150 Obj.	Gap	Test on N=200 Obj.	Gap
TSP	Ours (after training on 60)	7.8419	1.00%	9.6460	3.21%	11.2975	5.90%
	Ours (after training on 70)	7.8065	0.54%	9.5426	2.10%	11.1413	4.43%
	Ours (after training on 80)	7.7902	0.33%	9.4715	1.34%	11.0082	3.19%
	Ours (after training on 90)	7.7846	0.26%	9.4327	0.93%	10.9116	2.28%
	Ours (after training on 100)	7.7808	0.21%	9.4137	0.72%	10.8585	1.78%
	Ours (after training on 110)	7.7773	0.16%	9.3982	0.56%	10.8029	1.26%
	Ours (after training on 120)	7.7775	0.17%	9.3943	0.52%	10.7880	1.12%
	Ours (after training on 130)	7.7755	0.14%	9.3861	0.43%	10.7629	0.89%
	Ours (after training on 140)	7.7757	0.14%	9.3824	0.39%	10.7526	0.79%
	Ours (after training on 150)	7.7764	0.15%	9.3820	0.38%	10.7445	0.71%
CVRP	Ours (after training on 60)	15.9992	2.23%	20.0320	4.22%	23.5642	6.08%
	Ours (after training on 70)	15.8721	1.42%	19.6862	2.42%	23.0514	3.77%
	Ours (after training on 80)	15.8275	1.13%	19.5619	1.78%	22.8025	2.65%
	Ours (after training on 90)	15.8092	1.02%	19.5051	1.48%	22.7007	2.19%
	Ours (after training on 100)	15.7987	0.95%	19.4732	1.31%	22.6235	1.84%
	Ours (after training on 110)	15.7924	0.91%	19.4466	1.18%	22.5745	1.62%
	Ours (after training on 120)	15.7870	0.88%	19.4282	1.08%	22.5361	1.45%
	Ours (after training on 130)	15.7849	0.86%	19.4148	1.01%	22.4948	1.26%
	Ours (after training on 140)	15.7835	0.85%	19.4075	0.97%	22.4695	1.15%
	Ours (after training on 150)	15.7781	0.82%	19.3938	0.90%	22.4436	1.03%

D Validation on both seen and unseen sizes

We evaluate the obtained models after training on each size with intra-task regularization scheme. On both seen and unseen sizes, i.e., 100, 150 and 200, these models are evaluated to showcase that our approach can consistently improve the performance of the deep model, as the training progresses. The results are summarized in Table 8, where the gaps are calculated based on the solutions acquired by Concorde for TSP and HGS for CVRP in Table 1. We observe that when we evaluate the models on size 100, it progressively reduces the gaps before the training process reaching size 100. Furthermore, it effectively retains the acquired knowledge and superiority on size 100, after further trained on instances with larger sizes. The models trained after size 100 still show decreasing gaps on size 100. Similarly, in the case of testing on the larger sizes, i.e., 150 and 200, our approach consistently enhances the performance during the whole training phase. These findings highlight that our approach excels not only in preserving superior performance on small sizes but also in consistently enhancing performance on larger sizes.