

# 000 001 002 003 004 005 006 007 X-PLANES: ADAPTIVE AND EFFICIENT REPRESENTA- 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 054 055 056 057 058 059 060 061 062 063 064 065 066 067 068 069 070 071 072 073 074 075 076 077 078 079 080 081 082 083 084 085 086 087 088 089 090 091 092 093 094 095 096 097 098 099 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 444 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485 486 487 488 489 490 491 492 493 494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524 525 526 527 528 529 530 531 532 533 534 535 536 537 538 539 540 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555 556 557 558 559 559 560 561 562 563 564 565 566 567 568 569 569 570 571 572 573 574 575 576 577 578 579 579 580 581 582 583 584 585 586 587 588 589 589 590 591 592 593 594 595 596 597 598 599 599 600 601 602 603 604 605 606 607 608 609 609 610 611 612 613 614 615 616 617 618 619 619 620 621 622 623 624 625 626 627 628 629 629 630 631 632 633 634 635 636 637 638 639 639 640 641 642 643 644 645 646 647 648 649 649 650 651 652 653 654 655 656 657 658 659 659 660 661 662 663 664 665 666 667 668 669 669 670 671 672 673 674 675 676 677 678 679 679 680 681 682 683 684 685 686 687 688 689 689 690 691 692 693 694 695 696 697 698 699 699 700 701 702 703 704 705 706 707 708 709 709 710 711 712 713 714 715 716 717 718 719 719 720 721 722 723 724 725 726 727 728 729 729 730 731 732 733 734 735 736 737 738 739 739 740 741 742 743 744 745 746 747 748 749 749 750 751 752 753 754 755 756 757 758 759 759 760 761 762 763 764 765 766 767 768 769 769 770 771 772 773 774 775 776 777 778 779 779 780 781 782 783 784 785 786 787 788 789 789 790 791 792 793 794 795 796 797 798 799 799 800 801 802 803 804 805 806 807 808 809 809 810 811 812 813 814 815 816 817 818 819 819 820 821 822 823 824 825 826 827 828 829 829 830 831 832 833 834 835 836 837 838 839 839 840 841 842 843 844 845 846 847 848 849 849 850 851 852 853 854 855 856 857 858 859 859 860 861 862 863 864 865 866 867 868 869 869 870 871 872 873 874 875 876 877 878 879 879 880 881 882 883 884 885 886 887 888 889 889 890 891 892 893 894 895 896 897 898 899 899 900 901 902 903 904 905 906 907 908 909 909 910 911 912 913 914 915 916 917 918 919 919 920 921 922 923 924 925 926 927 928 929 929 930 931 932 933 934 935 936 937 938 939 939 940 941 942 943 944 945 946 947 948 949 949 950 951 952 953 954 955 956 957 958 959 959 960 961 962 963 964 965 966 967 968 969 969 970 971 972 973 974 975 976 977 978 979 979 980 981 982 983 984 985 986 987 988 989 989 990 991 992 993 994 995 996 997 998 999 1000 1001 1002 1003 1004 1005 1006 1007 1008 1009 1009 1010 1011 1012 1013 1014 1015 1016 1017 1018 1019 1019 1020 1021 1022 1023 1024 1025 1026 1027 1028 1029 1029 1030 1031 1032 1033 1034 1035 1036 1037 1038 1039 1039 1040 1041 1042 1043 1044 1045 1046 1047 1048 1049 1049 1050 1051 1052 1053 1054 1055 1056 1057 1058 1059 1059 1060 1061 1062 1063 1064 1065 1066 1067 1068 1069 1069 1070 1071 1072 1073 1074 1075 1076 1077 1078 1079 1079 1080 1081 1082 1083 1084 1085 1086 1087 1088 1089 1089 1090 1091 1092 1093 1094 1095 1096 1097 1098 1099 1099 1100 1101 1102 1103 1104 1105 1106 1107 1108 1109 1109 1110 1111 1112 1113 1114 1115 1116 1117 1118 1119 1119 1120 1121 1122 1123 1124 1125 1126 1127 1128 1129 1129 1130 1131 1132 1133 1134 1135 1136 1137 1138 1139 1139 1140 1141 1142 1143 1144 1145 1146 1147 1148 1149 1149 1150 1151 1152 1153 1154 1155 1156 1157 1158 1159 1159 1160 1161 1162 1163 1164 1165 1166 1167 1168 1169 1169 1170 1171 1172 1173 1174 1175 1176 1177 1178 1179 1179 1180 1181 1182 1183 1184 1185 1186 1187 1188 1189 1189 1190 1191 1192 1193 1194 1195 1196 1197 1198 1199 1199 1200 1201 1202 1203 1204 1205 1206 1207 1208 1209 1209 1210 1211 1212 1213 1214 1215 1216 1217 1218 1219 1219 1220 1221 1222 1223 1224 1225 1226 1227 1228 1229 1229 1230 1231 1232 1233 1234 1235 1236 1237 1238 1239 1239 1240 1241 1242 1243 1244 1245 1246 1247 1248 1249 1249 1250 1251 1252 1253 1254 1255 1256 1257 1258 1259 1259 1260 1261 1262 1263 1264 1265 1266 1267 1268 1269 1269 1270 1271 1272 1273 1274 1275 1276 1277 1278 1279 1279 1280 1281 1282 1283 1284 1285 1286 1287 1288 1289 1289 1290 1291 1292 1293 1294 1295 1296 1297 1298 1299 1299 1300 1301 1302 1303 1304 1305 1306 1307 1308 1309 1309 1310 1311 1312 1313 1314 1315 1316 1317 1318 1319 1319 1320 1321 1322 1323 1324 1325 1326 1327 1328 1329 1329 1330 1331 1332 1333 1334 1335 1336 1337 1338 1339 1339 1340 1341 1342 1343 1344 1345 1346 1347 1348 1349 1349 1350 1351 1352 1353 1354 1355 1356 1357 1358 1359 1359 1360 1361 1362 1363 1364 1365 1366 1367 1368 1369 1369 1370 1371 1372 1373 1374 1375 1376 1377 1378 1379 1379 1380 1381 1382 1383 1384 1385 1386 1387 1388 1389 1389 1390 1391 1392 1393 1394 1395 1396 1397 1398 1399 1399 1400 1401 1402 1403 1404 1405 1406 1407 1408 1409 1409 1410 1411 1412 1413 1414 1415 1416 1417 1418 1419 1419 1420 1421 1422 1423 1424 1425 1426 1427 1428 1429 1429 1430 1431 1432 1433 1434 1435 1436 1437 1438 1439 1439 1440 1441 1442 1443 1444 1445 1446 1447 1448 1449 1449 1450 1451 1452 1453 1454 1455 1456 1457 1458 1459 1459 1460 1461 1462 1463 1464 1465 1466 1467 1468 1469 1469 1470 1471 1472 1473 1474 1475 1476 1477 1478 1479 1479 1480 1481 1482 1483 1484 1485 1486 1487 1488 1489 1489 1490 1491 1492 1493 1494 1495 1496 1497 1498 1499 1499 1500 1501 1502 1503 1504 1505 1506 1507 1508 1509 1509 1510 1511 1512 1513 1514 1515 1516 1517 1518 1519 1519 1520 1521 1522 1523 1524 1525 1526 1527 1528 1529 1529 1530 1531 1532 1533 1534 1535 1536 1537 1538 1539 1539 1540 1541 1542 1543 1544 1545 1546 1547 1548 1549 1549 1550 1551 1552 1553 1554 1555 1556 1557 1558 1559 1559 1560 1561 1562 1563 1564 1565 1566 1567 1568 1569 1569 1570 1571 1572 1573 1574 1575 1576 1577 1578 1579 1579 1580 1581 1582 1583 1584 1585 1586 1587 1588 1589 1589 1590 1591 1592 1593 1594 1595 1596 1597 1598 1599 1599 1600 1601 1602 1603 1604 1605 1606 1607 1608 1609 1609 1610 1611 1612 1613 1614 1615 1616 1617 1618 1619 1619 1620 1621 1622 1623 1624 1625 1626 1627 1628 1629 1629 1630 1631 1632 1633 1634 1635 1636 1637 1638 1639 1639 1640 1641 1642 1643 1644 1645 1646 1647 1648 1649 1649 1650 1651 1652 1653 1654 1655 1656 1657 1658 1659 1659 1660 1661 1662 1663 1664 1665 1666 1667 1668 1669 1669 1670 1671 1672 1673 1674 1675 1676 1677 1678 1679 1679 1680 1681 1682 1683 1684 1685 1686 1687 1688 1689 1689 1690 1691 1692 1693 1694 1695 1696 1697 1698 1699 1699 1700 1701 1702 1703 1704 1705 1706 1707 1708 1709 1709 1710 1711 1712 1713 1714 1715 1716 1717 1718 1719 1719 1720 1721 1722 1723 1724 1725 1726 1727 1728 1729 1729 1730 1731 1732 1733 1734 1735 1736 1737 1738 1739 1739 1740 1741 1742 1743 1744 1745 1746 1747 1748 1749 1749 1750 1751 1752 1753 1754 1755 1756 1757 1758 1759 1759 1760 1761 1762 1763 1764 1765 1766 1767 1768 1769 1769 1770 1771 1772 1773 1774 1775 1776 1777 1778 1779 1779 1780 1781 1782 1783 1784 1785 1786 1787 1788 1789 1789 1790 1791 1792 1793 1794 1795 1796 1797 1798 1799 1799 1800 1801 1802 1803 1804 1805 1806 1807 1808 1809 1809 1810 1811 1812 1813 1814 1815 1816 1817 1818 1819 1819 1820 1821 1822 1823 1824 1825 1826 1827 1828 1829 1829 1830 1831 1832 1833 1834 1835 1836 1837 1838 1839 1839 1840 1841 1842 1843 1844 1845 1846 1847 1848 1849 1849 1850 1851 1852 1853 1854 1855 1856 1857 1858 1859 1859 1860 1861 1862 1863 1864 1865 1866 1867 1868 1869 1869 1870 1871 1872 1873 1874 1875 1876 1877 1878 1879 1879 1880 1881 1882 1883 1884 1885 1886 1887 1888 1889 1889 1890 1891 1892 1893 1894 1895 1896 1897 1898 1899 1899 1900 1901 1902 1903 1904 1905 1906 1907 1908 1909 1909 1910 1911 1912 1913 1914 1915 1916 1917 1918 1919 1919 1920 1921<br

054 Neural radiance fields (NeRF) has seen significant advancements since its introduction as a 3D  
 055 representation (Mildenhall et al., 2021). Considerable efforts have been made to extend NeRF to  
 056 dynamic scenes, which can be categorized into two main research lines: 1) incorporating time as an  
 057 additional input dimension to model a 4D NeRF representation (Fridovich-Keil et al., 2023; Shao  
 058 et al., 2023; Park et al., 2021a;b), and 2) disentangling a dynamic scene into a canonical radiance field  
 059 and a dynamic motion field (Pumarola et al., 2021; Liu et al., 2022; Fang et al., 2022a; Gan et al.,  
 060 2024). In NeRF-based methods, the current state-of-the-art (SOTA), such as  $K$ -Planes (Fridovich-Keil  
 061 et al., 2023) and Tensor4D (Shao et al., 2023), represent the 4D field using 6 or 9 planes, enabling both  
 062 a compact representation and efficient reconstruction of dynamic scenes. With recent developments  
 063 in large-scale pretraining for 3D generation (Hong et al., 2024; Zou et al., 2024; Tochilkin et al.,  
 064 2024), a natural question arises:

065 *What if we could start with an incomplete 4D field, rather than random initialization?*

066 The recently proposed Large Reconstruction Model (LRM) (Hong et al., 2024) takes an image as  
 067 input and predicts its 3D representation in the form of a triplane, presenting new opportunities for  
 068 triplane-based NeRF methods. For dynamic NeRF, LRM allows us to generate the triplanes for  
 069 each time frame from monocular video in just a few seconds. These triplanes can then be stacked  
 070 up to form an initial 4D field. However, this 4D field, especially in regions not captured by the  
 071 monocular video, may be noisy and incomplete. In this paper, we focus on how to derive an initial  
 072 compact representation, such as the 6/9 planes used in  $K$ -Planes (Fridovich-Keil et al., 2023) and  
 073 Tensor4D (Shao et al., 2023), by decomposing this noisy and incomplete 4D field. Besides that, the  
 074 pre-trained feature decoder (an MLP) in the LRM serves as the renderer (MLP) of the subsequent  
 075 dynamic NeRF, further optimized on the dynamic scene dataset. The decomposed fields with the  
 076 pre-trained renderer provide both a fast initialization and an effective regularization for subsequent  
 077 dynamic NeRF optimization, improving the SOTA dynamic NeRF methods (Fridovich-Keil et al.,  
 078 2023; Shao et al., 2023).

079 The existing NeRF-based tensor decomposition methods (Shao et al., 2023; Chen et al., 2022; Gao  
 080 et al., 2023) primarily provide a theoretical foundation for compact representations (e.g. vector  
 081 & matrix (Chen et al., 2022), 3 space planes (Chan et al., 2022), and 6 or 9 time-incorporated  
 082 planes (Fridovich-Keil et al., 2023; Shao et al., 2023)). These methods initialize compact repre-  
 083 sentations randomly and then directly optimize them in a data-driven manner, so the final features  
 084 are not directly tied to the initial fields. In contrast, our approach leverages a decomposition that  
 085 compresses the representation while preserving the quality of the initial 4D field. To achieve this, we  
 086 introduce Robust Principal Component Analysis (RPCA) for 4D field decomposition. As illustrated  
 087 in the lego bulldozer example (Figure 1), the static body and floor are low-rank components, while  
 088 the moving shovel is represented as sparse components. We reduce the low-rank results to 3 spatial  
 089 planes, which serve as a canonical space representation. For the sparse components, we treat them as  
 090 the supervision for guiding the training of space-time planes during the subsequent dynamic NeRF  
 091 optimization.

092 In summary, we propose  $X$ -Planes, a dynamic NeRF pipeline that integrates an effective initialization  
 093 method and an optimization strategy. By factorizing a 4D tensor derived from a large pretrained  
 094 model, the initialization method provides a compact and informative representation, along with  
 095 effective regularization for subsequent dynamic NeRF optimization. Experiments demonstrate the  
 096 method advantages of rendering performance, convergence speed, and robustness under the few-shot  
 097 settings. We achieve a PSNR of 20 within 30 seconds, making the optimization 2 $\times$  faster than  
 098 the baselines, and the improvements brought by  $X$ -Planes increases as the number of input views  
 099 decreases. Additionally, performance improves with larger LRM model sizes, suggesting the potential  
 100 to benefit from future advancements in LRM. These two key properties open the potential for broader  
 101 applicability in generalizable dynamic NeRF tasks.

## 103 2 RELATED WORKS

104  
 105  
 106 We first overview two research directions in dynamic NeRF, then briefly discuss Gaussian Splatting  
 107 for dynamic scene reconstruction. Finally, we review Large Reconstruction Model (LRM) and its  
 108 related methods.

108 In this section, we follow the formulation and notation from Yi et al. (2023) and present the standard  
 109 form of NeRF as:

$$p \xrightarrow{\text{MLP}_\theta} (c, \sigma), \quad (1)$$

110 where an MLP parameterized by  $\theta$  maps spatial positions  $p \in \mathbb{R}^3$  to RGB colors  $c \in [0, 1]^3$  and  
 111 density  $\sigma \in \mathbb{R}_{\geq 0}$ .  
 112

113 **Dynamic NeRF: 4D Field.** Representing dynamic objects with NeRF can be achieved in a straight-  
 114 forward manner by conditioning the original NeRF on the timestamp (Park et al., 2021b;a; Li et al.,  
 115 2022; 2021). This approach treats time  $t$  as an additional input dimension, using parameterization  
 116 method similar to static NeRF:

$$p, t \xrightarrow{\text{MLP}_\theta} (c, \sigma). \quad (2)$$

117 Subsequent methods (Fridovich-Keil et al., 2023; Shao et al., 2023; Cao and Johnson, 2023) utilize  
 118 compact feature planes to model the 4D field. These methods, grounded in tensor decomposition  
 119 theory, significantly accelerate training with less memory usage:

$$p, t \xrightarrow{\text{VT}_\Phi} Z \xrightarrow{\text{MLP}_\theta} (c, \sigma), \quad (3)$$

120 where the operation VT interpolates the compact feature planes  $\Phi$  to produce a latent feature  $Z \in \mathbb{R}^d$ ,  
 121 which is then decoded into standard radiance field outputs by an MLP with parameters  $\theta$ . As summa-  
 122 rized in Yi et al. (2023), VT typically involves three steps: (i) (Proj) **projecting** input coordinates  
 123 onto each feature plane, (ii) (Interp $_\Phi$ ) **interpolating** feature vectors from the corresponding planes  
 124  $\Phi$ , and (iii) (Reduce) **reducing** these interpolated features (e.g., concatenation, multiplication, or  
 125 addition) to generate the final latent representation  $Z$ . The process can be formulated as:

$$Z = \text{VT}_\Phi(p, t) = \text{Reduce}(\text{Interp}_\Phi(\text{Proj}(p, t))). \quad (4)$$

126 *In this paper, we build upon this research direction, further pushing the limits of performance and  
 127 training speed with large pretrained 3D reconstruction model LRM.*

128 **Dynamic NeRF: Canonical 3D Field + Motion Field.** To improve the disentanglement of shape  
 129 and motion, another research line proposes to model the 4D field as a combination of a canonical  
 130 3D field and a motion field. The pioneering work (Pumarola et al., 2021) introduces a deformable  
 131 neural radiance field that adopts a canonical 3D representation with 4D flow fields. Many subsequent  
 132 methods focus on accelerating static NeRF by using explicit data structures such as feature maps,  
 133 voxels and tensors. DeVRF (Liu et al., 2022) enables fast non-rigid neural rendering by combining  
 134 3D volumetric and 4D voxel fields (Song et al., 2023). V4D (Gan et al., 2024) introduces an effective  
 135 conditional positional encoding for 4D data, enabling fast novel view synthesis. Additionally,  
 136 TiNeuVox (Fang et al., 2022b) utilizes optimizable explicit data structures to accelerate radiance field  
 137 modeling.

138 **4D Gaussian Splatting.** Some recent works (Xu et al., 2024a; Wu et al., 2024) utilize 3D Gaussian  
 139 Splatting (3DGS) (Kerbl et al., 2023) to accelerate dynamic scene modeling. Building upon the  
 140 canonical 3D field with motion field approach, these methods replace the 3D NeRF voxel field with  
 141 3DGS to model the canonical 3D field. 3DGS is significantly faster than NeRF under sufficient input  
 142 views, but struggles with limited input views with challenges such as failed initialization, overfitting  
 143 on input images, and lack of details as detailed in (Yu et al., 2024). *In real-world dynamic scenes,  
 144 it is often challenging to capture enough views for a given scene. As a result, research on dynamic  
 145 NeRF with limited views remains valuable.*

146 **Large Reconstruction Model (LRM).** The recently proposed LRM (Hong et al., 2024) and its  
 147 follow-up works (Hong et al., 2024; Li et al., 2024a; Xu et al., 2024b; Wang et al., 2024) leverage a  
 148 pre-trained vision model, such as DINO (Caron et al., 2021), to encode the input image, and employ  
 149 a large transformer-based architecture to learn 3D representations of objects from a single image in a  
 150 data-driven manner. Such architecture consists of two functional components: an encoder  $f_\omega$  and a  
 151 decoder  $\text{MLP}_\theta$ . Their functionalities are formulated as follows:

$$I \xrightarrow{f_\omega} \Phi, \quad p \xrightarrow{\text{VT}_\Phi} Z \xrightarrow{\text{MLP}_\theta} (c, \sigma), \quad (5)$$

152 where  $f_\omega$  is the encoder that predicts the triplane  $\Phi$  from a single image. We can then synthesize any  
 153 novel views with  $\Phi$ , the VT operation as defined in Eq.(4), and the generalizable decoder  $\text{MLP}_\theta$ .

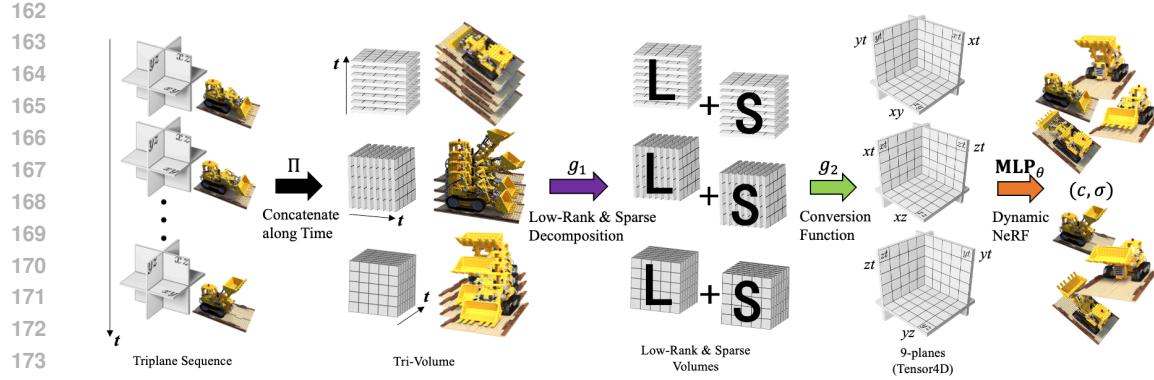


Figure 2: **X-Planes pipeline overview.** We omit the LRM  $f_\omega$  and start from the output triplane sequence  $\{\Phi_t\}_{t=1}^T$ . We first reorganize them into three time-aware 4D tensors by chunking and concatenating operator  $\Pi$ , then get the final feature planes  $\Phi$  through low-rank and sparse decomposition  $g_1$  and conversion function  $g_2$ . The feature planes  $\Phi$  and LRM’s NeRF decoder  $\text{MLP}_\theta$  are used to initialize that in dynamic NeRF methods such as *K*-Planes (Fridovich-Keil et al., 2023) and Tensor4D (Shao et al., 2023).

### 3 X-PLANES

In this section, we present **X**-Planes, a pipeline that includes an initialization method followed by a dynamic NeRF optimization strategy. We begin by formulating the pipeline in Sec. 3.1. Next, Sec. 3.2 introduces the low-rank and sparse decomposition strategy, and the reduction technique used in the initialization method. Finally, we discuss the dynamic NeRF optimization strategy in Sec. 3.3.

#### 3.1 FORMULATION

Given a set of images  $\{I_1^1, \dots, I_1^K, \dots, I_T^1, \dots, I_T^K\}$  captured over  $T$  timestamps of a dynamic scene with  $K$  fixed cameras, we select the  $k$ -th camera as the canonical pose. Using the images  $\{I_1^k, \dots, I_T^k\}$  across all timestamps, a time-series of triplanes  $\{\Phi_1, \dots, \Phi_T\}$  can be predicted using the LRM encoder  $f_\omega$ , where each triplane  $\Phi_t \in \mathbb{R}^{3 \times H \times W \times D}$ . The obtained triplanes  $\{\Phi_1, \dots, \Phi_T\}$  form an initial 4D field, which may be noisy and incomplete. Figure 2 illustrates how we decompose and reduce the 4D triplane sequence into compact feature planes. We formally define the entire initialization method of **X**-Planes as follows, where the decomposition and reduction procedure is denoted as  $\psi$ , with further details provided in section 3.2:

$$\{I_1^k, \dots, I_T^k\} \xrightarrow{f_\omega} \{\Phi_1, \dots, \Phi_T\} \xrightarrow{\psi} \Phi. \quad (6)$$

Once we obtain the decomposed feature planes  $\Phi$ , they are used to initialize the feature plane-based dynamic NeRF methods (Fridovich-Keil et al., 2023; Shao et al., 2023; Cao and Johnson, 2023). These methods can then be efficiently trained following their respective settings with an introduced regularization strategy, formulated as follows:

$$p, t \xrightarrow{\text{VT}_\Phi} Z \xrightarrow{\text{MLP}_\theta} (c, \sigma). \quad (7)$$

#### 3.2 TRIPLANE SEQUENCE TO COMPACT FEATURE PLANES

As illustrated in Figure 2, given a time-series sequence of triplanes  $\{\Phi_1, \dots, \Phi_T\} \in \mathbb{R}^{3 \times T \times H \times W \times D}$  obtained from LRM, we first reorganize them into three time-aware 4D tensors by chunking along the spatial dimension and concatenating along the time dimension:

$$\Pi(\Phi_1, \dots, \Phi_T) = \{\Phi_{xyt}, \Phi_{yzt}, \Phi_{xzt}\}, \quad (8)$$

where  $\Phi_{xyt}, \Phi_{yzt}, \Phi_{xzt} \in \mathbb{R}^{T \times H \times W \times D}$ , and  $\Pi$  is the composite of chunking and concatenating operations. The three time-aware 4D tensors can be regarded as three distinct views in the feature space, stacked along the time axis. Our objective is to derive the final feature planes

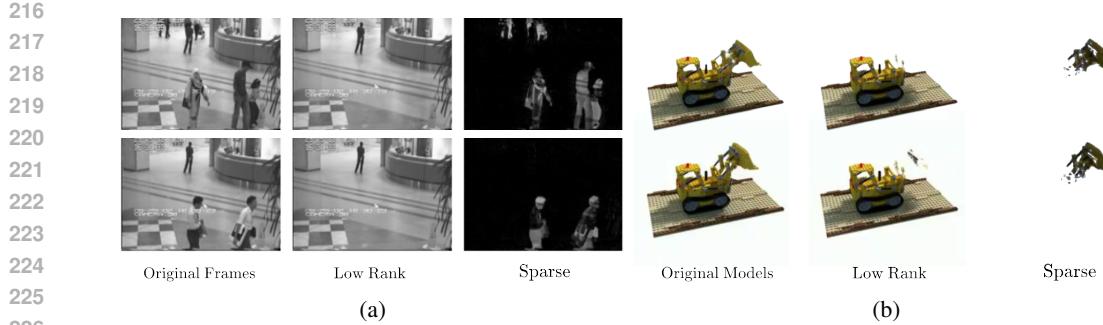


Figure 3: **Low-rank & sparse decomposition examples.** (a) Background and static objects are reserved in the low-rank, and moving pedestrians are extracted in the sparse. (b) The static components of the bulldozer remain in the low-rank, and the moving shovel are decomposed into the sparse.

$\Phi = \{\dots, \mathbf{P}_{xy}, \mathbf{P}_{yz}, \mathbf{P}_{xz}, \dots\}$  ( $\mathbf{P}_{\{\cdot\}} \in \mathbb{R}^{H \times W \times D}$ ) through low-rank and sparse decomposition function  $g_1$  and a conversion function  $g_2$  which transforms the decomposed components into optimizable feature planes. The overall transformation is represented as  $\psi = g_2 \circ g_1$ . Taking the 9 feature plane representation from Tensor4D (Shao et al., 2023) as an example, the process can be summarized as follows:

$$\begin{aligned} \Phi_{xyt} &\xrightarrow{g_1} \{\hat{\mathbf{L}}_{xyt}, \hat{\mathbf{S}}_{xyt}\} \xrightarrow{g_2} \{\mathbf{P}_{xy}, \mathbf{P}_{xt}, \mathbf{P}_{yt}\} \\ \Phi_{yzt} &\xrightarrow{g_1} \{\hat{\mathbf{L}}_{yzt}, \hat{\mathbf{S}}_{yzt}\} \xrightarrow{g_2} \{\mathbf{P}_{yz}, \mathbf{P}_{yt}, \mathbf{P}_{zt}\} \\ \Phi_{xzt} &\xrightarrow{g_1} \{\hat{\mathbf{L}}_{xzt}, \hat{\mathbf{S}}_{xzt}\} \xrightarrow{g_2} \{\mathbf{P}_{xz}, \mathbf{P}_{xt}, \mathbf{P}_{zt}\} \end{aligned} \quad (9)$$

The 6-plane representation from K-Plane (Fridovich-Keil et al., 2023) can be seen as a special case of the 9-plane representation, where feature planes with the same subscripts share parameters. The respective advantages and disadvantages of the 6-plane and 9-plane representations are discussed in the original papers.

**Low-Rank and Sparse Decomposition  $g_1$ .** The low rank and sparse decomposition approach is motivated by the successful application of robust PCA in video surveillance (Wright and Ma, 2022). Given a sequence of surveillance video frames, we often aim to identify dynamic activities against a static background. Stacking video frames into a matrix  $\mathbf{Y} \in \mathbb{R}^{(H \times W) \times T}$ , the low-rank component  $\mathbf{L} \in \mathbb{R}^{(H \times W) \times T}$  represents the stationary background, while the sparse component  $\mathbf{S} \in \mathbb{R}^{(H \times W) \times T}$  captures the moving objects in the foreground. The decomposed low-rank component  $\hat{\mathbf{L}}$  and sparse component  $\hat{\mathbf{S}}$  can be obtained by solving the following optimization problem:

$$\begin{aligned} \min \quad & \|\mathbf{L}\|_F + \lambda \|\mathbf{S}\|_1 \\ \text{s.t.} \quad & \mathbf{L} + \mathbf{S} = \mathbf{Y}, \end{aligned} \quad (10)$$

where  $\|\mathbf{L}\|_F$  denotes the Frobenius norm, promoting low-rank structure in  $\mathbf{L}$ , and  $\|\mathbf{S}\|_1$  is the  $l_1$ -norm encouraging sparsity in  $\mathbf{S}$ . The parameter  $\lambda$  balances the contribution of the low-rank and sparse components. This optimization can be efficiently solved using the Alternating Directions Method of Multipliers (ADMM) (Wright and Ma, 2022). The detailed procedure can be found in [Appendix](#) section. A specific example and its decomposition results are illustrated in Figure 3.

To apply robust PCA to the three time-aware 4D tensors in Eq.(8), we first reshape them to matrices  $\hat{\Phi}_{xyt}, \hat{\Phi}_{yzt}, \hat{\Phi}_{xzt} \in \mathbb{R}^{(H \times W \times D) \times T^1}$ . The reformulated optimization objective, adapted from Eq.(10), is expressed as:

$$\begin{aligned} \min \quad & \|\mathbf{L}\|_F + \lambda \|\mathbf{S}\|_1 \\ \text{s.t.} \quad & \mathbf{L} + \mathbf{S} = \hat{\Phi}_c, \quad c \in \{xyt, yzt, xzt\}. \end{aligned} \quad (11)$$

The final Lagrange function, formulated using the method of Lagrange multiplier, is:

$$\text{argmin}_{\mathbf{L}, \mathbf{S}} \quad \|\mathbf{L}\|_F + \lambda \|\mathbf{S}\|_1 + \frac{\mu}{2} \|\mathbf{L} + \mathbf{S} - \hat{\Phi}\|_2^2, \quad (12)$$

<sup>1</sup>To avoid ambiguity, all variable with  $\hat{\cdot}$  ( $\hat{\Phi}, \hat{\mathbf{L}}, \hat{\mathbf{S}}$ ) indicate 2D matrices and  $\tilde{\cdot}$  ( $\tilde{\Phi}, \tilde{\mathbf{L}}, \tilde{\mathbf{S}}$ ) indicates 4D tensors.

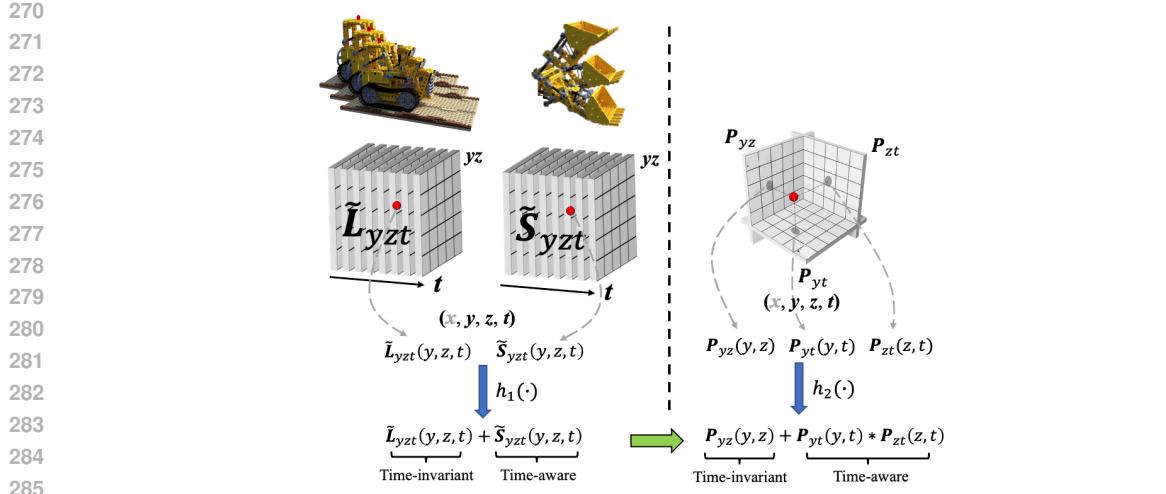


Figure 4: **Illustration of conversion function  $g_2$ .** Taking  $(y, z, t)$  as an example, the conversion function factorizes the feature voxels of the low-rank and sparse into planes.

which can be solved efficiently using ADMM. By applying  $\hat{\Phi}_{xyt}, \hat{\Phi}_{yzt}, \hat{\Phi}_{xzt}$  to Eq.(12) independently, we obtain three pairs of low-rank and sparse matrices  $(\hat{L}_{xyt}, \hat{S}_{xyt}), (\hat{L}_{yzt}, \hat{S}_{yzt}), (\hat{L}_{xzt}, \hat{S}_{xzt})$ . Each of these 6 matrices has the same dimension,  $\mathbb{R}^{(H \times W \times D) \times T}$ . To facilitate subsequent mathematical derivations, these matrices are reshaped back into 4D tensors:  $(\tilde{L}_{xyt}, \tilde{S}_{xyt}), (\tilde{L}_{yzt}, \tilde{S}_{yzt}), (\tilde{L}_{xzt}, \tilde{S}_{xzt})$ , with dimensions  $\mathbb{R}^{H \times W \times T \times D}$ .

**Conversion function  $g_2$ .** In this paragraph, we discuss how to transform the decomposed low-rank and sparse components into optimizable feature planes. Before that, we must clarify the connection between the decomposed low-rank and sparse 4D tensors and compact feature planes.

As shown in Figure 4, suppose we have a pair of low-rank & sparse 4D tensors and 3 feature planes, sharing the same decoder. Both of them can utilize this decoder to render an image at arbitrary viewpoints and timestamps. To render a given point  $(x, y, z)$  at the  $t$ -th timestamp, we first need to sample the corresponding features: feature pair  $\{\tilde{L}_{yzt}(y, z, t), \tilde{S}_{yzt}(y, z, t)\}$  from the low-rank & sparse 4D tensors, and triplet features  $\{P_{yz}(y, z), P_{yt}(y, t), P_{zt}(z, t)\}$  from the 3 feature planes. The key challenge is to determine the combination functions,  $h_1(\cdot)$  and  $h_2(\cdot)$ , that satisfy the subsequent equations:

$$h_1(\tilde{L}_{yzt}(y, z, t), \tilde{S}_{yzt}(y, z, t)) = h_2(P_{yz}(y, z), P_{yt}(y, t), P_{zt}(z, t)). \quad (13)$$

The second assumption is that the shared decoder is identical to the NeRF decoder used in LRM. Under this assumption,  $h_1(\cdot)$  can be defined as an additive operation, following the constraint from Eq.(11), which ensures that the low-rank and sparse matrices sum back to the original feature  $\hat{\Phi}$  which can be directly rendered by the LRM’s NeRF decoder. Furthermore, as illustrated in Figure 3, the low-rank component is time-invariant, implying that the sparse component must be time-aware. Consequently,  $h_2(\cdot)$  also becomes an additive operator that combines time-invariant and time-aware components. In the triplet features  $\{P_{yz}(y, z), P_{yt}(y, t), P_{zt}(z, t)\}$ , only  $P_{yz}(y, z)$  is time-invariant, while  $P_{yt}(y, t), P_{zt}(z, t)$  are time-aware. Following the approach from (Fridovich-Keil et al., 2023), we multiply the time-aware components to obtain the desired features. This leads to the final forms of  $h_1(\cdot)$  and  $h_2(\cdot)$ , as depicted in Figure 4<sup>2</sup>. Finally, we adopt the feature merging strategy of LRM for sampled features from the planes. These features are concatenated and subsequently fed into the MLP decoder. This process yields the concatenated low-rank and sparse features for  $(x, y, z, t)$ :

$$\begin{bmatrix} \Phi_{xyt}(x, y, t) \\ \Phi_{yzt}(y, z, t) \\ \Phi_{xzt}(x, z, t) \end{bmatrix} = \begin{bmatrix} \tilde{L}_{xyt}(x, y, t) \\ \tilde{L}_{yzt}(y, z, t) \\ \tilde{L}_{xzt}(x, z, t) \end{bmatrix} + \begin{bmatrix} \tilde{S}_{xyt}(x, y, t) \\ \tilde{S}_{yzt}(y, z, t) \\ \tilde{S}_{xzt}(x, z, t) \end{bmatrix}, \quad (14)$$

<sup>2</sup>For simplicity, we take the  $yzt$  part as an example. The  $xyt$  and  $xzt$  parts share symmetric operations.

324 and the corresponding features from feature planes:  
 325

$$326 \quad \begin{bmatrix} \mathbf{P}_{xy}(x, y) \\ \mathbf{P}_{yz}(y, z) \\ \mathbf{P}_{xz}(x, z) \end{bmatrix} + \begin{bmatrix} \mathbf{P}_{xt}(x, t) \cdot \mathbf{P}_{yt}(y, t) \\ \mathbf{P}_{yt}(y, t) \cdot \mathbf{P}_{zt}(z, t) \\ \mathbf{P}_{xt}(x, t) \cdot \mathbf{P}_{zt}(z, t) \end{bmatrix}. \quad (15)$$

330 Straightforwardly, we can use the right side of Eq.(14) to supervise Eq.(15) during dynamic NeRF  
 331 training. However, as shown in Figure 3, the decomposed low-rank matrices hold the time-invariant  
 332 information, satisfying the requirements for  $\mathbf{P}_{xy}$ ,  $\mathbf{P}_{yz}$ ,  $\mathbf{P}_{xz}$  in Eq.(9). From a visual perspective, the  
 333 decomposed low-rank matrices  $\hat{\mathbf{L}}$  is indistinguishable along the time axis. Therefore, we can simply  
 334 choose the chunk of  $\hat{\mathbf{L}}$  corresponding to the  $\frac{T}{2}$  timestamp to **initialize** the time-invariant feature  
 335 planes:

$$336 \quad \mathbf{P}_{xy} \doteq \tilde{\mathbf{L}}_{xyt} \left( \frac{T}{2} \right), \quad \mathbf{P}_{yz} \doteq \tilde{\mathbf{L}}_{yzt} \left( \frac{T}{2} \right), \quad \mathbf{P}_{xz} \doteq \tilde{\mathbf{L}}_{xzt} \left( \frac{T}{2} \right), \quad (16)$$

339 where  $\tilde{\mathbf{L}}_{xyt}(\frac{T}{2})$  indicates selection along the time axis of the 4D tensor  $\tilde{\mathbf{L}}_{xyt}$  corresponding to the  
 340  $\frac{T}{2}$ -th timestamp. The ablation study in the experiment section shows that the average rank of low-rank  
 341 matrices is 2.53 which also aligns with the conclusions of our analysis.

### 343 3.3 DYNAMIC NERF OPTIMIZATION

344 **Feature Decoders  $\mathbf{MLP}_\theta$ .** We use the pretrained NeRF MLP decoder from LRM as our feature  
 345 decoder. The MLP is comprised of multiple linear layers with ReLU activations. The output is  
 346 a 4-dimensional vector, where the first three dimensions represent the RGB colors, and the last  
 347 dimension corresponds to the density of the field.

349 **Losses.** In addition to the regularization terms introduced in (Fridovich-Keil et al., 2023; Shao et al.,  
 350 2023), we incorporate the sparse component  $\tilde{\mathbf{S}}$  to regularize the time-aware feature planes  $\mathbf{P}_S$  with  
 351 loss:

$$353 \quad \mathcal{L}_r = \|\tilde{\mathbf{S}} - \mathbf{P}_S\|_2^2, \quad (17)$$

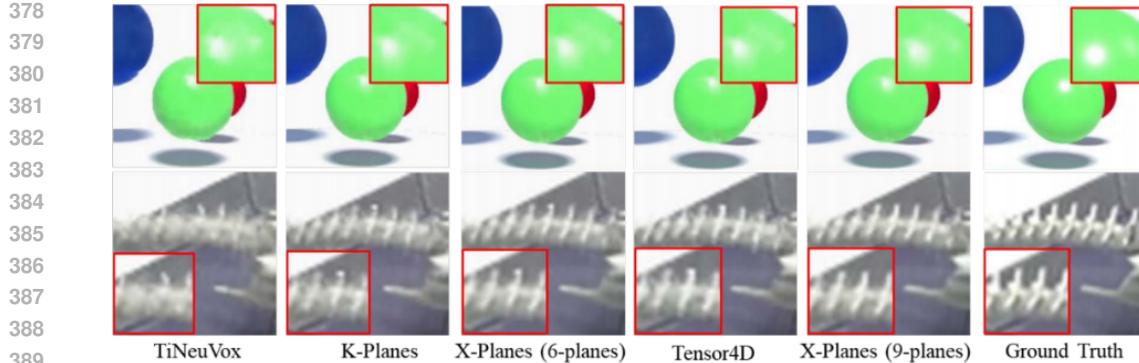
355 where  $\tilde{\mathbf{S}}$  is the last term in Eq.(14) and  $\mathbf{P}_S$  is the second term in Eq.(15), namely:

$$357 \quad \tilde{\mathbf{S}} = \begin{bmatrix} \tilde{\mathbf{S}}_{xyt}(x, y, t) \\ \tilde{\mathbf{S}}_{yzt}(y, z, t) \\ \tilde{\mathbf{S}}_{xzt}(x, z, t) \end{bmatrix}, \quad \mathbf{P}_S = \begin{bmatrix} \mathbf{P}_{xt}(x, t) * \mathbf{P}_{yt}(y, t) \\ \mathbf{P}_{yt}(y, t) * \mathbf{P}_{zt}(z, t) \\ \mathbf{P}_{xt}(x, t) * \mathbf{P}_{zt}(z, t) \end{bmatrix}.$$

## 361 4 EXPERIMENTS

363 **Dataset.** The D-NeRF dataset (Pumarola et al., 2021) contains eight scenes of varying duration (50  
 364 to 200 frames). Each timestamp provides a single training image from a different viewpoint. In our  
 365 setting, we require a monocular video with a fixed viewpoint as input to the LRM to generate the  
 366 initial triplanes. Hence, we train K-Planes for the eight scenes and render a time-series of fixed-view  
 367 images. For evaluation, standardized test views are taken from novel camera positions at various  
 368 timestamps throughout the video.

370 **Baselines.** We compare our method against the state-of-the-art (SOTA) from two research directions  
 371 in dynamic NeRF, Tensor4D (Shao et al., 2023) and K-Planes (Fridovich-Keil et al., 2023), which  
 372 directly learn time-conditioned 4D fields. For fair comparison, we retrain them in our experiments,  
 373 achieving improved results compared to their original papers. On the other hand, TiNeuVox (Fang  
 374 et al., 2022b) is the SOTA methods from the other direction: modelling a canonical 3D field with  
 375 an associated motion field. Both methods rely on explicit voxel grids to represent the canonical 3D  
 376 field. Apart from the NeRF-based methods, we also compare our method with the gaussian splatting  
 377 methods, including 4D-GS (Wu et al., 2024), Spacetime GS (Li et al., 2024b), and Saro-GS (Yan  
 et al., 2024).

389  
390  
391  
392  
393  
394  
395  
396  
397  
398  
399  
400  
401  
402  
403  
404  
405  
406  
407  
408  
409  
410  
411  
412  
413  
414  
415  
416  
417  
418  
419  
420  
421  
422  
423  
424  
425  
426  
427  
428  
429  
430  
431  
Figure 5: Qualitative comparison with SOTA methods.

**Implementation Details.** For robust PCA, we set  $\mu = 0.05$  and  $\lambda = 0.009$  for the ADMM optimization. We borrow the LRM architecture as the same as that in the openLRM Github repository. In our experiments, we evaluate  $X$ -Planes with two configurations: 6-planes and 9-planes. The ablation study on different LRM versions of varying parameter sizes, multi-resolution configurations, and the 6/9-plane representations of  $X$ -Planes is provided in the [Appendix](#) section. All the experiments are trained on a single NVIDIA 3090 GPU.

400  
401  
402  
403  
404  
405  
406  
407  
408  
409  
410  
Table 1: Quantitative comparison on D-NeRF dataset.

Genre	Name	PSNR $\uparrow$	SSIM $\downarrow$	Time $\downarrow$
GS	4D-GS	34.09	0.981	24 min
	Spacetime GS	36.07	0.984	50 min
	Saro-GS	36.13	0.985	50 min
NeRF	$K$ -Planes	31.71	0.971	52 min
	Tensor4D	32.33	0.973	64 min
	TiNeuVox	32.67	0.973	28 min
	$X$ -Planes (9 Planes)	33.67	0.980	34 min
	$X$ -Planes (6 Planes)	33.40	0.979	26 min

411  
412  
413  
414  
415  
416  
417  
418  
419  
420  
421  
422  
423  
424  
425  
426  
427  
428  
429  
430  
431  
Table 2: Quantitative comparison on different LRM sizes.

Size	PSNR $\uparrow$	SSIM $\downarrow$
Small	31.13	0.966
Base	31.78	0.970
Large	32.18	0.973
xLarge	33.40	0.979

## 4.1 PERFORMANCE COMPARISON

**Comparison on reconstruction accuracy.** We train our model for each individual scene and provide example results for novel view synthesis in Figure 5. The quantitative results in Table 1 demonstrates that our method outperforms NeRF-based SOTA methods in both rendering quality and reconstruction speed. With 2 $\times$  training speed,  $X$ -Planes achieves 1.21 PSNR improvement over the SOTA method Tensor4D, and 2.7 PSNR over  $K$ -Planes using similar 6-plane representation.  $X$ -Planes also produce competitive results compared with gaussian splatting methods.

**Comparison on few training iterations.** We further compare  $X$ -Planes with baseline methods under a limited number of training iterations to highlight the effectiveness of our  $X$ -Planes initialization method. As shown in Figure 6,  $X$ -Planes achieves relatively good visual results with just 200 optimization iterations. This demonstrates the potential of  $X$ -Planes for fast deployment in applications where speed is critical and slight compromises in quality are acceptable.

## 4.2 FEW-SHOT AND DIFFERENT LRM MODELS

**Effects of different LRM model sizes.** In an ideal scenario, the LRM can perfectly reconstruct a static scene. Using our  $X$ -Planes initialization method, this enables the direct construction of a compact 4D field with feature planes, capable of rendering high-quality images at arbitrary viewpoints and timestamps. Consequently, the effectiveness and efficiency of  $X$ -Planes should improve as LRM

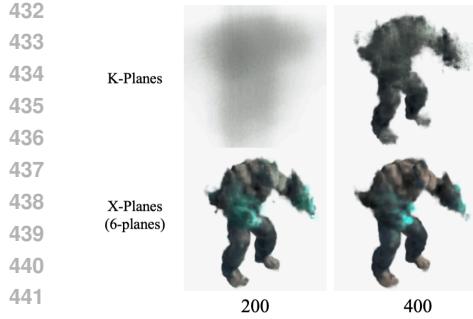


Figure 6: **Comparison with random initialization.**  $X$ -Planes’ training converges faster than the vanilla. Bottom numbers indicate iterations.

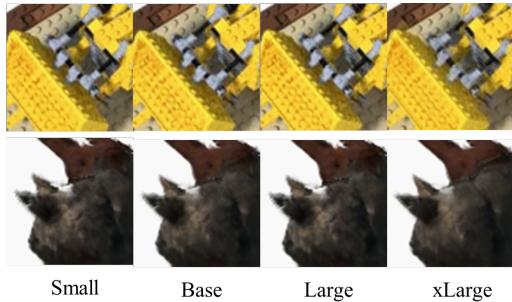


Figure 7: **Qualitative comparison on different model size of LRM.** As the LRM size increases, the reconstruction result quality improves.

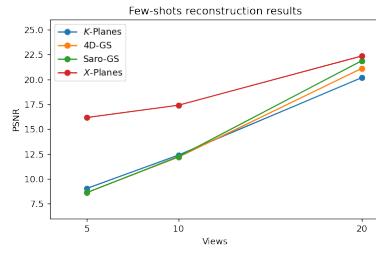


Figure 8: **Quantitative comparison of few-shots reconstruction.**

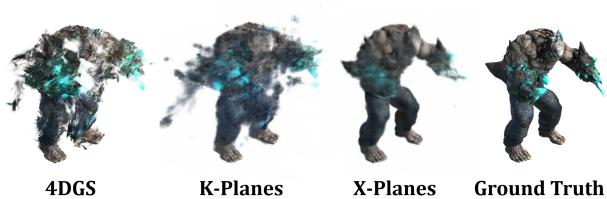


Figure 9: **Qualitative comparison of models under few-shots settings.** The number of input views is 5.

456 performance increases, highlighting the importance of an effective initialization strategy. Table 2 and  
457 Figure 7 provide both quantitative and qualitative evidences showing that  $X$ -Planes benefits from  
458 enhanced LRM performance, where LRM models of varying sizes are used to approximate models  
459 with varying capabilities. We borrow different LRM model sizes from Github Repositories<sup>34</sup>. Small,  
460 Base, and xLarge in Table.2 correspond to the models small, base, and large from the OpenLRM  
461 repository. Large in Table 2 refers to the model from the TripoSR repository.

462 **Few-shot results.** Figure 8 shows the PSNR and qualitative results of  $K$ -Planes, SaRo-GS (Yan  
463 et al., 2024), 4DGs (Wu et al., 2024) and  $X$ -Planes at different few-shot levels (5 views, 10 views,  
464 and 20 views). Figure 9 shows the corresponding visualized results. As the # of views decreases,  
465 the advantage of  $X$ -Planes over the other methods increases. Experiments prove that given a few  
466 input images, the initialization and regularization of the LRM-generated planes make the model more  
467 robust to overfitting. This result, cooperates with that in different LRM model sizes, opening the  
468 potential for broader applicability in generalizable dynamic NeRF tasks.

## 5 CONCLUSION

473 This paper proposes  $X$ -Planes, a novel dynamic NeRF pipeline with an effective initialization and  
474 optimization strategy. It employs a pretrained Large Reconstruction Model (LRM) to generate an  
475 initial noisy and incomplete 4D field of a dynamic scene. This initial representation is subsequently  
476 decomposed into compact feature planes through low-rank and sparse decomposition. The resulting  
477 decomposed feature planes serve as both an effective initialization and a form of regularization for  
478 subsequent dynamic NeRF optimization, enabling new state-of-the-art results demonstrated through  
479 qualitative and quantitative results. Additionally,  $X$ -Planes empowers fast deployment in applications  
480 where speed is prioritized over absolute quality. Furthermore,  $X$ -Planes is broadly applicable to  
481 dynamic NeRF methods, ready to benefit from future advancements of LRM, paving the way for  
482 more generalizable dynamic NeRF tasks.

<sup>3</sup><https://github.com/3DTopia/OpenLRM>

<sup>4</sup><https://github.com/VAST-AI-Research/TripoSR>

486 REFERENCES  
487

488 Sara Fridovich-Keil, Giacomo Meanti, Frederik Rahbæk Warburg, Benjamin Recht, and Angjoo  
489 Kanazawa. K-planes: Explicit radiance fields in space, time, and appearance. In *Proceedings of the*  
490 *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12479–12488,  
491 June 2023.

492 Ruizhi Shao, Zerong Zheng, Hanzhang Tu, Boning Liu, Hongwen Zhang, and Yebin Liu. Tensor4d:  
493 Efficient neural 4d decomposition for high-fidelity dynamic reconstruction and rendering. In  
494 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*,  
495 pages 16632–16642, June 2023.

496 T. Kanade, P. Rander, and P.J. Narayanan. Virtualized reality: constructing virtual worlds from real  
497 scenes. *IEEE MultiMedia*, 4(1):34–47, 1997.

498

499 Jonathan Starck and Adrian Hilton. Surface capture for performance-based animation. *IEEE*  
500 *Computer Graphics and Applications*, 27(3):21–31, 2007.

501 Alvaro Collet, Ming Chuang, Pat Sweeney, Don Gillett, Dennis Evseev, David Calabrese, Hugues  
502 Hoppe, Adam Kirk, and Steve Sullivan. High-quality streamable free-viewpoint video. *ACM Trans.*  
503 *Graph.*, 34(4), July 2015. ISSN 0730-0301.

504

505 Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and  
506 Ren Ng. Nerf: representing scenes as neural radiance fields for view synthesis. *Commun. ACM*, 65  
507 (1):99–106, December 2021. ISSN 0001-0782.

508

509 Keunhong Park, Utkarsh Sinha, Jonathan T. Barron, Sofien Bouaziz, Dan B Goldman, Steven M.  
510 Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. In *Proceedings of*  
511 *the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5865–5874, October  
512 2021a.

513 Keunhong Park, Utkarsh Sinha, Peter Hedman, Jonathan T. Barron, Sofien Bouaziz, Dan B Goldman,  
514 Ricardo Martin-Brualla, and Steven M. Seitz. Hypernerf: a higher-dimensional representation for  
515 topologically varying neural radiance fields. *ACM Trans. Graph.*, 40(6), December 2021b. ISSN  
516 0730-0301.

517 Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural  
518 radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer*  
519 *Vision and Pattern Recognition (CVPR)*, pages 10318–10327, June 2021.

520

521 Jia-Wei Liu, Yan-Pei Cao, Weijia Mao, Wenqiao Zhang, David Junhao Zhang, Jussi Keppo, Ying  
522 Shan, Xiaohu Qie, and Mike Zheng Shou. Devrf: Fast deformable voxel radiance fields for  
523 dynamic scenes. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors,  
524 *Advances in Neural Information Processing Systems*, volume 35, pages 36762–36775. Curran  
525 Associates, Inc., 2022.

526

527 Jiemin Fang, Taoran Yi, Xinggang Wang, Lingxi Xie, Xiaopeng Zhang, Wenyu Liu, Matthias Nießner,  
528 and Qi Tian. Fast dynamic radiance fields with time-aware neural voxels. In *SIGGRAPH Asia 2022*  
529 *Conference Papers*, SA ’22, New York, NY, USA, 2022a. Association for Computing Machinery.  
ISBN 9781450394703.

530

531 Wanshui Gan, Hongbin Xu, Yi Huang, Shifeng Chen, and Naoto Yokoya. V4d: Voxel for 4d novel  
532 view synthesis. *IEEE Transactions on Visualization and Computer Graphics*, 30(2):1579–1591,  
533 2024.

534

535 Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli,  
536 Trung Bui, and Hao Tan. LRM: Large reconstruction model for single image to 3d. In *The Twelfth*  
537 *International Conference on Learning Representations*, 2024.

538

539 Zi-Xin Zou, Zhipeng Yu, Yuan-Chen Guo, Yangguang Li, Ding Liang, Yan-Pei Cao, and Song-Hai  
Zhang. Triplane meets gaussian splatting: Fast and generalizable single-view 3d reconstruction  
with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern*  
*Recognition (CVPR)*, pages 10324–10335, June 2024.

540 Dmitry Tochilkin, David Pankratz, ZeXiang Liu, Zixuan Huang, Adam Letts, Yangguang Li, Ding  
 541 Liang, Christian Laforte, Varun Jampani, and Yan-Pei Cao. Tripocr: Fast 3d object reconstruction  
 542 from a single image. *CoRR*, abs/2403.02151, 2024.

543

544 Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields.  
 545 In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner,  
 546 editors, *Computer Vision – ECCV 2022*, pages 333–350, Cham, 2022. Springer Nature Switzerland.  
 547 ISBN 978-3-031-19824-3.

548

549 Quankai Gao, Qiangeng Xu, Hao Su, Ulrich Neumann, and Zexiang Xu. Strivec: Sparse tri-vector  
 550 radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision  
 551 (ICCV)*, pages 17569–17579, October 2023.

552

553 Eric R. Chan, Connor Z. Lin, Matthew A. Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio  
 554 Gallo, Leonidas J. Guibas, Jonathan Tremblay, Sameh Khamis, Tero Karras, and Gordon Wetzstein.  
 555 Efficient geometry-aware 3d generative adversarial networks. In *Proceedings of the IEEE/CVF  
 556 Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16123–16133, June 2022.

557

558 Brent Yi, Weijia Zeng, Sam Buchanan, and Yi Ma. Canonical factors for hybrid neural fields.  
 559 In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages  
 3414–3426, October 2023.

560

561 Tianye Li, Mira Slavcheva, Michael Zollhöfer, Simon Green, Christoph Lassner, Changil Kim, Tanner  
 562 Schmidt, Steven Lovegrove, Michael Goesele, Richard Newcombe, and Zhaoyang Lv. Neural 3d  
 563 video synthesis from multi-view video. In *Proceedings of the IEEE/CVF Conference on Computer  
 564 Vision and Pattern Recognition (CVPR)*, pages 5521–5531, June 2022.

565

566 Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time  
 567 view synthesis of dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer  
 568 Vision and Pattern Recognition (CVPR)*, pages 6498–6508, June 2021.

569

570 Ang Cao and Justin Johnson. Hexplane: A fast representation for dynamic scenes. In *Proceedings of  
 571 the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 130–141,  
 June 2023.

572

573 Liangchen Song, Anpei Chen, Zhong Li, Zhang Chen, Lele Chen, Junsong Yuan, Yi Xu, and Andreas  
 574 Geiger. Nerfplayer: A streamable dynamic scene representation with decomposed neural radiance  
 575 fields. *IEEE Transactions on Visualization and Computer Graphics*, 29(5):2732–2742, 2023.

576

577 Jiemin Fang, Taoran Yi, Xinggang Wang, Lingxi Xie, Xiaopeng Zhang, Wenyu Liu, Matthias Nießner,  
 578 and Qi Tian. Fast dynamic radiance fields with time-aware neural voxels. In *SIGGRAPH Asia 2022  
 579 Conference Papers*, SA '22, New York, NY, USA, 2022b. Association for Computing Machinery.  
 ISBN 9781450394703.

580

581 Zhen Xu, Sida Peng, Haotong Lin, Guangzhao He, Jiaming Sun, Yujun Shen, Hujun Bao, and Xiaowei  
 582 Zhou. 4k4d: Real-time 4d view synthesis at 4k resolution. In *Proceedings of the IEEE/CVF  
 583 Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20029–20040, June  
 2024a.

584

585 Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian,  
 586 and Xinggang Wang. 4d gaussian splatting for real-time dynamic scene rendering. In *Proceedings  
 587 of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20310–  
 20320, June 2024.

588

589 Bernhard Kerbl, Georgios Kopanas, Thomas Leimkuehler, and George Drettakis. 3d gaussian  
 590 splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4), July 2023. ISSN  
 591 0730-0301.

592

593 Hanyang Yu, Xiaoxiao Long, and Ping Tan. Lm-gaussian: Boost sparse-view 3d gaussian splatting  
 with large model priors, 2024.

594 Jiahao Li, Hao Tan, Kai Zhang, Zexiang Xu, Fujun Luan, Yinghao Xu, Yicong Hong, Kalyan  
 595 Sunkavalli, Greg Shakhnarovich, and Sai Bi. Instant3d: Fast text-to-3d with sparse-view gen-  
 596 eration and large reconstruction model. In *The Twelfth International Conference on Learning  
 597 Representations*, 2024a.

598 Yinghao Xu, Hao Tan, Fujun Luan, Sai Bi, Peng Wang, Jiahao Li, Zifan Shi, Kalyan Sunkavalli,  
 599 Gordon Wetzstein, Zexiang Xu, and Kai Zhang. DMV3d: Denoising multi-view diffusion using 3d  
 600 large reconstruction model. In *The Twelfth International Conference on Learning Representations*,  
 601 2024b.

602 Peng Wang, Hao Tan, Sai Bi, Yinghao Xu, Fujun Luan, Kalyan Sunkavalli, Wenping Wang, Zexiang  
 603 Xu, and Kai Zhang. PF-LRM: Pose-free large reconstruction model for joint pose and shape  
 604 prediction. In *The Twelfth International Conference on Learning Representations*, 2024.

605 Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and  
 606 Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of  
 607 the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9650–9660, October  
 608 2021.

609 John Wright and Yi Ma. *High-Dimensional Data Analysis with Low-Dimensional Models: Principles,  
 610 Computation, and Applications*. Cambridge University Press, 2022.

611 Zhan Li, Zhang Chen, Zhong Li, and Yi Xu. Spacetime gaussian feature splatting for real-time  
 612 dynamic view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and  
 613 Pattern Recognition (CVPR)*, pages 8508–8520, June 2024b.

614 Jinbo Yan, Rui Peng, Luyang Tang, and Ronggang Wang. 4d gaussian splatting with scale-aware  
 615 residual field and adaptive optimization for real-time rendering of temporally complex dynamic  
 616 scenes. In *Proceedings of the 32nd ACM International Conference on Multimedia*, MM '24,  
 617 page 7871–7880, New York, NY, USA, 2024. Association for Computing Machinery. ISBN  
 618 9798400706868.

619  
 620  
 621  
 622  
 623  
 624  
 625  
 626  
 627  
 628  
 629  
 630  
 631  
 632  
 633  
 634  
 635  
 636  
 637  
 638  
 639  
 640  
 641  
 642  
 643  
 644  
 645  
 646  
 647

648 APPENDICES  
649650 ALTERNATING DIRECTIONS METHOD OF MULTIPLIERS (ADMM)  
651652 In this section, we concretize how the low-rank and sparse decomposition, i.e., the robust PCA, is  
653 achieved by the Alternating Directions Method of Multipliers (ADMM) algorithm, which solves the  
654 optimization objective below,

655 
$$\begin{aligned} \min & \quad \|\mathbf{L}\|_F + \lambda\|\mathbf{S}\|_1 \\ 656 \text{s.t.} & \quad \mathbf{L} + \mathbf{S} = \hat{\Phi}, \end{aligned} \tag{18}$$

658 where  $\hat{\Phi}$  is the known feature matrix (refer to Eq.(18) the of the main body), and  $\mathbf{L}$  and  $\mathbf{S}$  are the  
659 unknown low-rank and sparse components, respectively. A generic Lagrange multiplier algorithm  
660 would optimize the augmented Lagrangian, adapted from Eq.(19),  
661

662 
$$\mathcal{L}_\mu(\mathbf{L}, \mathbf{S}, \Lambda) = \|\mathbf{L}\|_F + \lambda\|\mathbf{S}\|_1 + \langle \Lambda, \Delta \rangle + \frac{\mu}{2}\|\Delta\|_2^2, \tag{19}$$

663 where  $\langle \cdot, \cdot \rangle$  denotes the matrix inner product, and  $\Delta$  denotes the residual matrix term,  
664

665 
$$\Delta = \mathbf{L} + \mathbf{S} - \hat{\Phi}. \tag{20}$$

666 With the lagrange multiplier  $\Lambda$  initialized as  $\mathbf{0}$ , the Lagrange multiplier algorithm would repeatedly  
667 update

668 
$$\begin{aligned} (\mathbf{L}_{k+1}, \mathbf{S}_{k+1}) &= \arg \min_{\mathbf{L}, \mathbf{S}} \mathcal{L}_\mu(\mathbf{L}, \mathbf{S}, \Lambda_k), \\ 669 \Lambda_{k+1} &= \Lambda_k + \mu \Delta_{k+1}. \end{aligned} \tag{21}$$

670 To efficiently solve the optimization problem above, the ADMM algorithm recognizes it as two  
671 sub-problems:  $\min_{\mathbf{L}} \mathcal{L}_\mu(\mathbf{L}, \mathbf{S}, \Lambda)$  and  $\min_{\mathbf{S}} \mathcal{L}_\mu(\mathbf{L}, \mathbf{S}, \Lambda)$ .  
672673 The sparse component is minimized by  
674

675 
$$\arg \min_{\mathbf{S}} \mathcal{L}_\mu(\mathbf{L}, \mathbf{S}, \Lambda) = \mathcal{S}_{\lambda/\mu}(\hat{\Phi} - \mathbf{L} - \mu^{-1}\Lambda), \tag{22}$$

676 where  $\mathcal{S}_{\lambda/\mu}$  denotes the shrinkage operator. For a single element  $x$  in the matrix,  
677

678 
$$\mathcal{S}_{\lambda/\mu}[x] = \text{sgn}(x) \max(|x| - \lambda/\mu, 0), \tag{23}$$

679 and the operation extends to matrices in an element-wise way.  
680681 The low-rank component is minimized by  
682

683 
$$\arg \min_{\mathbf{L}} \mathcal{L}_\mu(\mathbf{L}, \mathbf{S}, \Lambda) = \mathcal{D}_{1/\mu}(\hat{\Phi} - \mathbf{S} - \mu^{-1}\Lambda). \tag{24}$$

684 Here,  $\mathcal{D}_{1/\mu}$  denotes the singular value thresholding operator,  
685

686 
$$\mathcal{D}_{1/\mu}(\mathbf{M}) = \mathbf{U} \mathcal{S}_{1/\mu}(\Sigma) \mathbf{V}^*, \tag{25}$$

687 where  $\mathbf{M} = \mathbf{U} \Sigma \mathbf{V}^*$  is the singular value decomposition.  
688689 The strategy of the ADMM algorithm is to first optimize  $\mathcal{L}_\mu$  with respect to  $\mathbf{L}$  (fixing  $\mathbf{S}$ ), then  
690 minimize  $\mathcal{L}_\mu$  with respect to  $\mathbf{S}$  (fixing  $\mathbf{L}$ ), and then finally update the Lagrange multiplier  $\Lambda$ . The  
691 process is summarized as **Algorithm 1**.  
692693 **Algorithm 1** Robust PCA by ADMM  
694

- 1: Initialize:  $\mathbf{S}_0 = \Lambda_0 = \mathbf{0}$ ,  $\mu > 0$ .
- 2: **while** not converged **do**
- 3:   Compute  $\mathbf{L}_{k+1} = \mathcal{D}_{1/\mu}(\hat{\Phi} - \mathbf{S}_k - \mu^{-1}\Lambda_k)$ ;
- 4:   Compute  $\mathbf{S}_{k+1} = \mathcal{S}_{\lambda/\mu}(\hat{\Phi} - \mathbf{L}_{k+1} - \mu^{-1}\Lambda_k)$ ;
- 5:   Compute  $\Lambda_{k+1} = \Lambda_k + \mu(\mathbf{L}_{k+1} + \mathbf{S}_{k+1} - \hat{\Phi})$ ;
- 6: **end while**
- 7: Output:  $\mathbf{L}_* \leftarrow \mathbf{L}_k$ ;  $\mathbf{S}_* \leftarrow \mathbf{S}_k$ .

702 ABLATION EXPERIMENTS ON LRM SIZES, RESOLUTIONS, AND NUMBERS OF PLANES  
703704 In this section, we experiment with our  $X$ -Planes on different LRM model sizes, multi-resolution,  
705 and 6/9-planes representations. With such ablation studies, we investigate the performance gains  
706 from various aspects of  $X$ -Planes respectively.707 **LRM size and 6/9-planes representation.** In Table. 3, we ablate our model with respect to different  
708 sizes of the LRM models and the number of feature planes. It shows the larger model size of LRM  
709 would directly increase the performance of  $X$ -Planes.711 Table 3: **Ablation study over LRM sizes and six/nine-planes representation.**  
712

Model Size	6-Planes PSNR $\uparrow$	9-Planes PSNR $\uparrow$	6-Planes SSIM $\uparrow$	9-Planes SSIM $\uparrow$
Small	31.13	32.04	0.969	0.972
Base	31.78	32.54	0.971	0.975
Large	32.18	32.54	0.972	0.974
xLarge	33.40	33.67	0.978	0.981

721 **Multi-resolution.** In Table 4, we use LRM model with xLarge size and ablate our model with  
722 respect to different resolutions of the feature plane and their impact on performance metrics. The  
723 results refer to a different conclusion from that in Tensor4D and K-Planes. In Tensor4D and K-Planes,  
724 the performance increases significantly with the increase of feature plane resolution, whereas it is not  
725 influenced much by feature plane resolution in our method. The MLP decoder architecture and its  
726 pretrained weights (from LRM) might be the main reason. We will explore it in our feature work.727 Table 4: **Ablation study over plane resolutions.**  
728

Plane Resolution $\uparrow$	6-Planes PSNR $\uparrow$	9-Planes PSNR $\uparrow$	6-Planes SSIM $\uparrow$	9-Planes SSIM $\uparrow$
64	33.40	33.56	0.979	0.980
128	33.51	33.65	0.980	0.981
256	33.55	33.68	0.981	0.981
512	33.55	33.67	0.981	0.981

## 736 MORE QUALITATIVE RESULTS

737 This section shows more cases of Section 4.1. (see Figure 10)

754 Figure 10: Qualitative comparison with SOTA methods.  
755